Study of Serial Line Protocols
And
Design of PPP-IP Interface

by

VINOD KAMAT

DEPARTMENT OF ELECTRICAL ENGINEERING

INDIAN INSTITUTE OF TECHNOLOGY KANPUR

APRIL, 1995

Study of Serial Line Protocols And Design of PPP-IP Interface

A Thesis Submitted in Partial
Fulfilment of the Requirements
for the Degree of
Master of Technology,
by

Vinod Kamat

Department of Electrical Engineering
Indian Institute of Technology Kanpur

April 1995

CERTIFICATE

It is certified that the work contained in the thesis titled Study of Serial Line Protocols And Design of PPP-IP Interface, by *Vinod Kamat* (Registration No. 9310459) has been carried out under my supervision and that this work has not been submitted elsewhere for a degree.

K R Srivathsan

April, 1995

Dr. K.R. Srivathsan,
Professor,
Department of Electrical Engineering,
IIT, Kanpur.

DEDICATED

TO

DOORDARSHAN

दूरदर्शन

सत्यम् शिवम् सुन्दरम्

Acknowledgements

I am deeply indebted to Dr. K. R. Srivatsan, my guide, for his invaluable guidance and the moral support. Without him I would have not been in a position to complete this work. I am thankful to Dr. S. K. Bose for introducing me to the world of computer networks. I am also thankful to Dr. D. Manjunath for his suggestions on the presentation of this thesis work.

I am greatful to this PC/AT 486 system on which most of the implementation and the typesetting work is carried out. I am thankful to Kaushal, Raju, Ramanath and Bhatnagerji. I am also greatful to the DOORDARSHAN LAB for providing the necessary facilities during the final formatting of the thesis document.

I thank Sanjeev and Sandhya for all their help and suggestions. I am thankful to Vinod Kumar, Navpreet, Amithab Roy, T. S. Rao, Narayan, Manish, Vivek and Uma for their help and cheerful company in the LAB. I am thankful to Deepak Murthy for his help during LATEX compilation.

I am extremely greatful to Girisha for all his help throughout his stay here. Without him some of the courses would have not been so simple. I am thankful to Bhavesh and Kirti for their help and nice company.

I am greatly indebted to Q-1 SBRA for providing me a very comfortable stay and a very facinating environment. I am thankful to all the defence friends and my colleagues at SBRA Quarters. The company of Prakash Veer, Manoj, Dash, Ranganadh, Kaali Prasad, Burman, Krishnamurty and Kumar is particularly memorable. I take this opportunity to thank all the members of *Kannada Sangha*, particularly Suresh, Praveen, Madhu, Sriharsha and Govindraj.

I am greatly indebted to *DOORDARSHAN*, Government of India, for providing me an opportunity to persue this masters programme at IIT, Kanpur. I thank all my colleagues at Bombay for extending the necessary cooperation throughout this programme.

It is with the deep sense of gratitude that I thank my parents and my brothers Murlianna and Ramakantanna. Without their support and encouragement it would have been impossible for me to come and work here.

27th April 1995 Vinod Kamat.

Abstract

Wide Area Networks (WANs) are formed by interconnection of several geographically distant, independently managed Local Area Networks (LANs) and hosts. Routers with serial links are often used to interconnect these LANs or hosts. The serial links are prone to errors and the data transmitted may be corrupted by noise. In order to provide reliable data transmission over the serial links, link control protocols are employed at the data link layer of the OSI model.

This thesis presents a study and comparison of some of the widely used serial line data link protocols such as HDLC, LAPB, LAPD, LAPF, PPP and SLIP. The thesis reviews in detail the evolution of LAPF, the link layer protocol for accessing the Frame relay network, its protocol structure and discusses the features available for congestion management. The design of a PPP-IP interface is discussed. An attempt has been made to enhance the existing PPP protocol implementation so as to enable the routers in use at IITK to handle the data arriving on the serial link.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

Wide Area Networks (WANs) often consist of a conglomeration of several geographically distant Local Area Networks (LANs) or host systems, interconnected together by serial data links. Local Area Networks enable a small group of users to communicate only among themselves. Wide Area Networks on the other hand enable the users in one group to communicate with those in the other. Public data networks provide one way for this wide area connectivity. The users are required to hand over the data to the public network and it is the responsibility of the network to deliver the data to the destination. These public data networks are accessed using the serial links connecting the user sites (the local area networks or hosts) to the network. Another way for wide area connectivity is for two distant LANs or hosts to be interconnected directly by a serial link without the intervention of the public network. Often a router connects a LAN to another LAN or to a remote host over this serial link. Thus a serial data link of one kind or another form an indispensable component of wide area networks. Problems with serial links include random bit errors and short outages. This makes the serial links unreliable for the data transmission. These problems are best tackled by a suitable serial line protocol operating at the data link layer of the OSI model [1]. There are several standard serial line protocols like SDLC,HDLC, LAPB, LAPD, LAPF and PPP in wide use today.

The IP-Routers used in the IITK campus LAN execute the Internet Protocol (IP) at the network layer and the Ethernet Protocol at the data link layer. But they do not provide any protocol at the data link layer for the reliable transmission of data on the serial line. As a result these routers can not be used for interconnecting geographically distant LANs or hosts. This led to the implementation of the point-to-point protocol (PPP) in the ERNET LAB [2]. However this implementation has several limitations. For instance, it can receive the data packets from a remote system only in the polled mode. Further the data exchange between the two systems is possible only at the data link level. The PPP is not able to exchange the data with the IP network layer above it and hence the router is not able to route data packets on serial links. In order to enable the router to route the data packets on the serial link, a suitable software interface is needed to bridge the gap between the PPP and the IP modules.

Further, activities in the area of fast packet technologies like ATM and Frame relay are in progress in the Telematics LAB. Under this a frame relay user-to-network interface is proposed to be designed and developed. In this regard it is necessary to study the standards, protocol structure and decide upon the software and hardware requirements.

The objectives of this thesis are ;

- to review the various serial line data link protocols,
- to carry out the pre-implementation studies of a frame relay user-to-network interface and
- to design and develop an interface module for splicing the PPP and IP modules.

1.2 Wide Area Networks: An Overview

Intercity, intercountry and intercontinental networks are known as the Wide Area Networks. Just as the need to share the information among desktop computers in an office has forced the proliferation of local area networks, the need to share the information beyond a single workgroup has forced the adoption of LAN-to-LAN or LAN-to-host interconnections using gateways, routers and other network systems. There are several media over which the information exchange can take place between two LANs, a LAN and a host or between two hosts. This section will provide an overview of some of the options available for achieving the information exchange between two geographically distant LANs or host systems.

Wide area networks commonly employ transmission media such as telephone lines, satellites, microwaves, traditional data networks and modems in order to facilitate the communication among the various user sites. In many of these categories the communication path is serial, normally provided by the telephone network. Figure 1.1 depicts a typical wide area communication scenario.

A Public Switched Telephone Network (PSTN) or a Public Switched Data Network (PSDN) often provides the interconnection of geographically distant LANs or host systems. For the transmission of digital data over the PSTN network, which basically handles analog signals, *modems* are used. Modems convert the digital data in to analog form at the transmitting end and vice versa at the receiving end. Modems are driven by a communication software package that runs on the terminal or the personal computer of the user. In the PSTN network, a connection may be established using a dial-up facility or a dedicated non-switched telephone line.

In the PSDN context a DTE communicates with a DCE on the data network side. The function of the data communication network is to interconnect the DTEs, via these DCEs, so that they can share resources, exchange data, and provide backup for each other.

As depicted in the figure, a host on LAN A may communicate with any host on LAN B via the PSTN network or the PSDN network. Alternately, they may communicate directly over the serial link connecting the two LANs. Similarly, the hosts such as A, B, C and D, which access the public networks directly, may communicate with each other or with any host on LAN A or LAN B via these public networks.

A DCE-DTE pair in the PSDN network or the two systems interconnected via the PSTN network communicate with *protocols*. The protocols are agreements on how the machines communicate with each other. Typically, several layers of protocols are required to support the information exchange between different hosts. The International Organization for Standards' (ISO's) Open

System Interconnection (OSI) model specifies a seven layer protocol architecture. The lowest two layers, *viz.* the physical and the data link, provide the basic point-to-point communication between the two systems involved in a serial line communications architecture. The Physical layer protocol specifies the electrical characteristics of the physical link between the two systems. The data link layer protocol provides the reliable communication between them using this unreliable physical link. Since the data path between the two systems is serial these data link protocols are generally termed as serial line protocols. The following section provides a brief overview of the serial line protocols.

1.3 Serial Line Protocols: An Overview

There are several serial line protocols defined by different international organizations like ISO and CCITT. Some of these protocols are noted below.

1. *Synchronous Data Link Control (SDLC) Protocol.* This is defined by IBM. The link may be either point-to-point or point-to-multipoint [3]. Both half- and full-duplex link operations are supported.

2. *High level Data Link Control (HDLC) protocol.* This is specified by the ISO [3] and is widely used. This protocol supports either point-to-point or point-multipoint mode of operation. The link may be configured for either half- or full-duplex operations.

3. *Link Access Protocol - Balanced (LAPB).* This is the link level protocol for the data exchange between an X.25 packet switched network and the system connected to it [4].

4. *Link Access Protocol - D Channel (LAPD).* This is the link level protocol for the data and control information exchange between the ISDN customer premises equipment and an ISDN network over the D channel [5].

5. *Link Access Protocol - Frame relay (LAPF).* This is the link level protocol for the data transfer between a frame relay system and the frame relay network [6].

6. *Point-to-Point Protocol (PPP).* This is the standard serial line protocol for the transmission of data units from multiple network layer protocols over the serial line between the two peers.

In other words it provides the multiplexing of multiple network layer data units over the single serial link [7]. Supports only full duplex link operation over both leased and switched circuits.

7. *BInary SYNChronous (BISYNC) Protocol.* This is defined by IBM exclusively for the half-duplex link operation [8]. It uses two *SYNC* characters in each frame, the data link layer transmission unit.

8. *MONO SYNChronous (MONOSYNC) Protocol.* This is a variation of the BISYNC protocol. This uses only one *SYNC* character in each frame [8].

9. *Serial Line IP (SLIP).* This is the serial line protocol for the transmission of only IP data units over the serial line between two peers. It is widely used due to its simplicity [9]. It does not support error detection, error correction, addressing, the barest minimum features a data communication system is expected to provide.

10. *XNS Synchronous point-to-point protocol.* This is the serial line data link protocol defined for the data transfer between remote LANs in *Xerox Network Systems (XNS)* Architecture [10]. Both half and full duplex operations are supported over both leased and switched circuits.

HDLC, LAPB, LAPD, LAPF, PPP and SLIP protocols are discussed in more detail in the subsequent chapters.

1.4 Outline Of This Thesis

The rest of this thesis is organized as follows.

The next chapter discusses the HDLC protocol. LAPB, LAPD and SLIP protocols are also discussed in here.

The point-to-point protocol (PPP) is discussed in the third chapter. It also discusses the Link Control Protocol (LCP) and the Internet Protocol Control Protocol (IPCP) used by PPP for its operation.

Chapter four introduces the frame relay technology. It discusses, the evolution, the protocol structure and the features available in LAPF for congestion management.

The design of PPP-IP interface is discussed in chapter five. Implementation of a polling scheme to enable the router to transact data packets on the serial links is also discussed.

The sixth chapter concludes this thesis with the presentation of the salient distinctions between the serial line protocols studied in the previous chapters and the suggestions for the future work.

Chapter 2

HDLC, LAPB, LAPD and SLIP Protocols

2.1 Introduction

The data transmission across the serial communication link must be error-free. Further it must flow in a controlled and orderly manner. But these communication links are prone to distortions and the data carried may be corrupted by noise. Due to this reason the serial links are somewhat unreliable. Serial line data link layer protocols are employed to deal with the errors that may occur and thereby provide reliable data transmission between the two systems interconnected by the link. The services provided by these protocols are enumerated below.

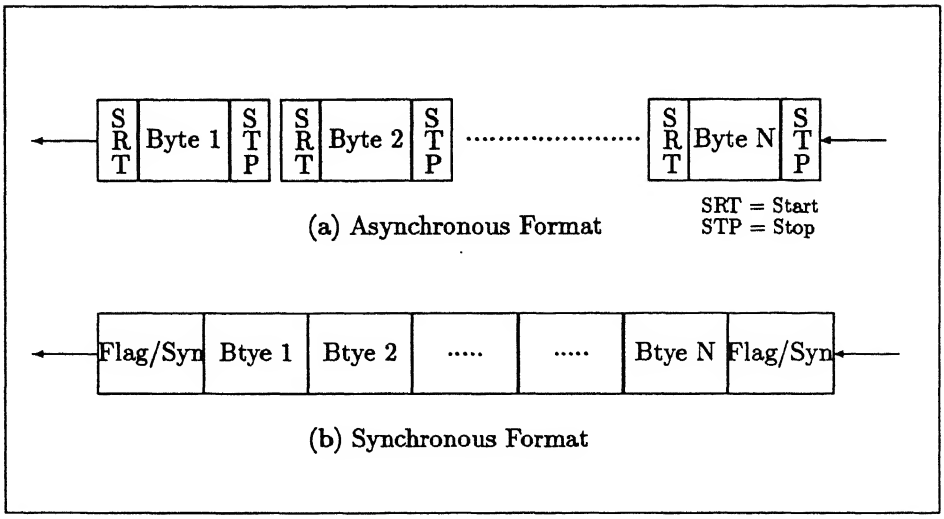
1. Synchronizing (logically not physically) the sender and receiver through the use of flags/SYNC characters.

2. Controlling the flow of data to prevent the sender from sending too fast.

3. Detecting and recovering from errors between two points on the link.

4. Distinguishing between data and control characters
   and

5. Determining the identity of the communicating stations.



(a) Asynchronous Format

(b) Synchronous Format

Figure 2.1: Asynchronous and Synchronous Transmissions

The data link layer rests above the physical layer in the OSI reference model. These two layers together provide an error-free communication link between the two systems thereby ensuring orderly and correct delivery of packets between them. A variety of serial line protocols have been specified for this purpose.

These protocols run over two kinds of physical devices; the asynchronous and the synchronous. Accordingly there are two types of data transmissions, asynchronous and synchronous. These are depicted in figure 2.1. In the asynchronous type the data are transmitted one character at a time i.e. the characters are sent independently of each other. Whereas in the synchronous type the data are transmitted in blocks of characters or bits.

With this brief introduction, we now discuss various serial line data link layer protocols such as HDLC, LAPB, LAPD and SLIP in the subsequent sections. PPP and LAPF are discussed in little more detail in chapters 3 and 4 respectively.

2.2 High level Data Link Control Protocol

This protocol is designed to operate over a single physical medium between two stations. It is a bit-oriented protocol [11] specified by the International Standards Organization (ISO) as ISO 3309 and ISO 4335. It is being very widely used. This protocol is based on the SDLC (Synchronous Data

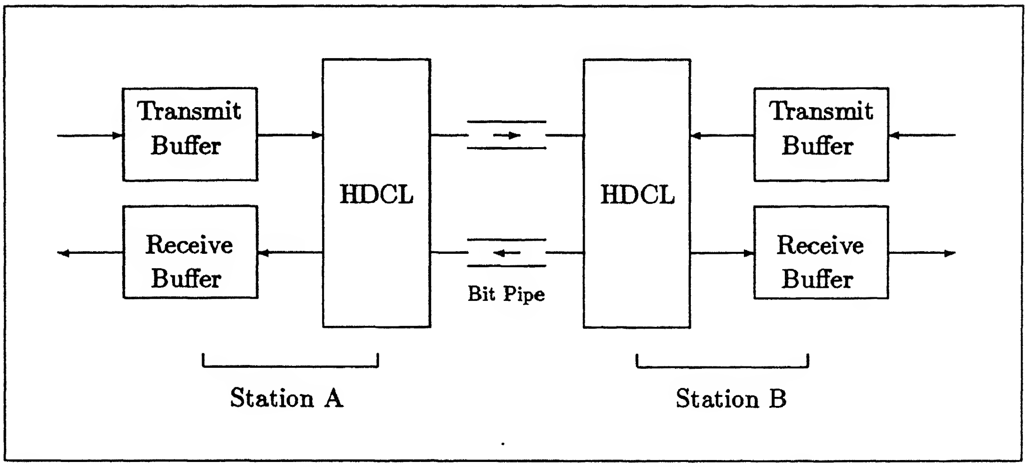


Figure 2.2: HDLC Data Link Model

Link Control) protocol specified by IBM. HDLC supports both half duplex and full duplex transmissions; point-to-point and point-to-multipoint configurations; and switched and non-switched channels.

HDLC is used as a basis for other protocols that use specific options defined in the HDLC repertoire. For instance, the CCITT X.25 recommended data link procedures, LAPB (Link Access Protocol - Balanced) is a subset of HDLC. Similarly, LAPD (Link Access Protocol - D Channel,ISDN), LAPF (Link Access Protocol - Frame relay) and PPP (Point-to-Point Protocol) are all derived from HDLC. For this reason HDLC is discussed in considerable detail in this chapter and deals with the basic principle and the operation.

Figure 2.2 gives the basic structure of an HDLC data link model. Stations A and B communicate in the full duplex mode. An HDLC station is classified as one of the following three types.

Primary Station

A primary station is the one which controls the data link. This station acts as a master and transmits command frames to the secondary stations on the channel. In turn, it receives response frames from those stations. If the link is multipoint, the primary station is responsible for maintaining a separate session with each station attached to the link.

Secondary Station

A secondary station acts as a slave to the primary station. It responds to the commands from the primary station in the form of responses. It maintains only one session, that being with the primary station, and has no responsibility for control on the link.

Combined Station

A combined station transmits both the commands and the responses and receives both the commands and the responses from another combined station.

Three modes of operation are defined for the HDLC protocol which are enumerated below.

2.2.1 Normal Response Mode (NRM)

In this mode the stations involved in communication are said to use unbalanced channel configuration [12] in which one primary station communicates with one or more secondary stations in Point-to-point or multipoint half duplex, full duplex, switched or non-switched mode. The primary station becomes the sole master of the channel and the secondary stations need to receive explicit permission from the primary before transmitting. The primary station grants permission to the secondary station(s) in a polled fashion. After receiving permission, the secondary station initiates a response transmission which may contain data. The transmission may consist of one or more frames while the channel is being used by the secondary station. NRM is used frequently on multipoint links.

2.2.2 Asynchronous Response Mode (ARM)

This mode is similar to NRM except that the secondary station does not need permission from the primary station to initiate transmission. This mode decreases the overhead since the secondary station does not need a poll sequence in order to send data. A secondary station operating in this mode can transmit only when it detects an idle channel state for a two way alternate (half duplex) data flow, or at any time for a two way simultaneous (full duplex) data flow. The primary station maintains responsibility for tasks such as error recovery, link setup and link disconnection.

2.2.3 Asynchronous Balanced Mode (ABM)

This mode is exclusively meant for Point-to-Point link transmissions. The communication takes place only between two stations interconnected over this link. In this mode both the stations involved in communication serve as equal partners. A class of procedures defined for this mode forms the basis for the link level procedures of the X.25 protocol. These are defined there as LAPB [13]. LAPB is discussed briefly in the next section.

2.2.4 HDLC Frame Structure

The standard frame structure for HDLC appears in the figure 2.3. The frame consists of four or five fields. The information field may not be present in some type of frames[1]. The address, control and frame check sequence (FCS) fields can all be increased to 2, 2, 4 octets respectively in order to allow for extended addressing, increased sequence numbers and improved error detection. The protocol recognizes the bit sequences of;

- at least seven, but less than fifteen 1s as *abort*

  and

- fifteen or more 1s as *idle line* .

The functions of each field in an HDLC frame are briefly discussed below.

Flag Field:

A unique sequence of 01111110 is used as flag in order to delimit the frame. Bit stuffing is used to eliminate the possibility of the flag sequence appearing in the data portion of the frame. A zero is inserted at the transmitter any time that five consecutive 1s appear outside the flag fields. The inserted zero is removed/destuffed at the receiver.

Control Field:

The control field can be one or two octets. The structure of this field decides the type of HDLC frame being transmitted. Three types of frames are defined to handle information flow, supervisory

[1] S and U type of frames do not carry any information field.

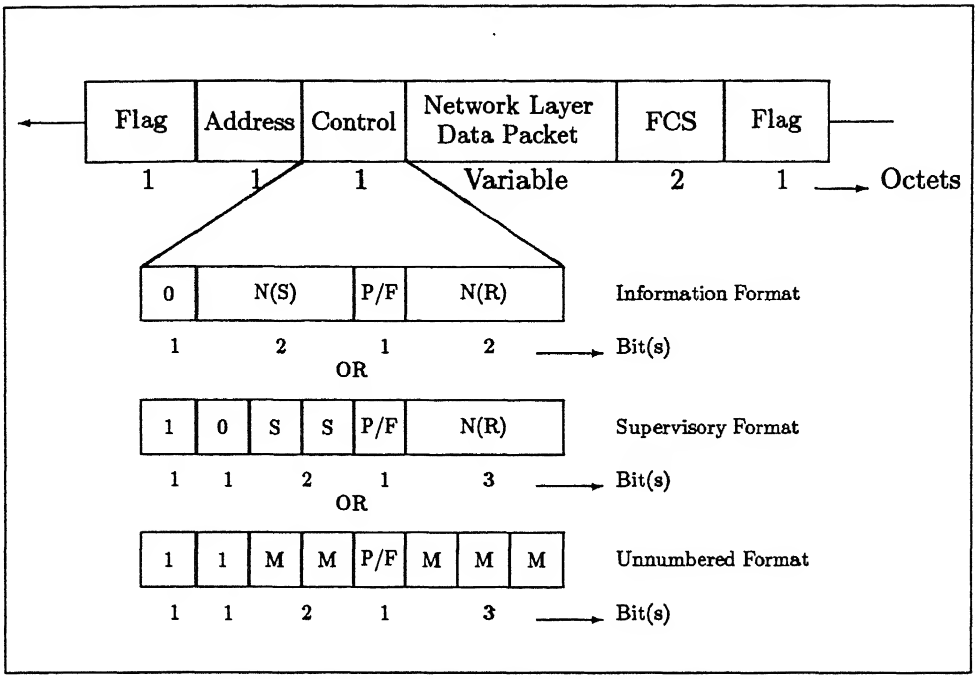


Figure 2.3: Standard HDLC Frame Structure

and control signals, and responses to all these. These frames are;

1. I (Information) format
2. S (Supervisory) format
3. U (Unnumbered) format

It may be noted that these three types of frames differ only in the control field format. A zero in the first bit of the control field corresponds to an I-frame. The bit pairs 10 and 11 appearing as the first two bits indicate an S-frame and a U-frame respectively. While the I-frames are intended for carrying the user data, S- and U-frames[2] do not carry any user data. They are used strictly for supervisory and control purposes. Thus the 8-bit control field in a frame determines the type of the HDLC frame being transmitted and further for S and U types of frames the specific signal being transmitted. The P/F bit decides whether a frame is a command or a response [2.2.5]. The sequence numbers N(S) and N(R) are used for the acknowledging undamaged frames and retransmission of damaged frames. They also facilitate *flow-controlling* [2.2.6]. The functions of individual frames are briefly discussed later in this chapter.

[2] Except for UI form of U-frames which carry user data in the information field [2.2.10].

Address Field:

The address field is one octet. It is useful only in point-to-multipoint or multidrop link configurations. It does not have any significance in point-to-point link configurations. In the unbalanced mode of HDLC (NRM and ARM), wherein we have one primary station communicating with one or more secondary stations the address is always that of the secondary station. In the balanced mode (ABM), wherein combined stations are involved the address is always that of the responding station. Since an I-frame is always a command the address must be that of receiving station. In case of S-frames, if the frame is a command the address is that of the receiving station and if a response the frames carry their own addresses.

Information Field:

The datagram arriving from the network layer is placed in this field for onward transmission. Only I-frames and UI (Unnumbered Information) form of U-frames carry this field. The maximum length of this field is not defined in the standards [1]. Often this is limited to the *maximum transfer unit* of the corresponding network layer.

Frame check Sequence:

This field has two octets and is used for error detection. CCITT-16 CRC code is normally used. FCS is calculated over all bits of *address, control,* and *information* fields. It does not include *flag* fields or the FCS field itself. Optionally this field can have four octets in which case CCITT-32 CRC code is used. This option is employed when the data bit sequence is too long and the link is too erroneous.

2.2.5 Commands/Responses

The frames may be command frames or response frames. HDLC imposes a restriction that I-frames be only command frames. The RR and RNR type of S-frames may be either commands or responses. REJ, SREJ type of S-frames are always responses. The U-frames can be commands or responses. The P/F bit in the control field decides whether a frame is a command or a response.

2.2.6 Error Control and flow Control

Error control is a technique wherein a receiving station requests the sender for the retransmission of the frames in error. And flow control is a technique for assuring that a transmitting station does not overwhelm a receiving station with data. Sliding-window is the most popularly used protocol for these purposes. The three bit number N(S) in the I-frame represents the sequence number of the I-frame. Mod-8 sequence numbering is thus standard with normal HDLC. Each successive I-frame has its sequence number increased by one. When the transmitter reaches its maximum sequence number it is forced to stop transmitting until a frame in the reverse direction is received, acknowledging an outstanding frame. The N(R) bits in the I- and S-frames are used to acknowledge I-frames received. The number N(R) acknowledges the receipt of N(R)-1 and any frames preceding that number, not already acknowledged. N(R) indicates that the receiver is expecting I-frame number N(R). The N(S) and N(R) fields can be extended to seven bits to allow Mod-128 sequence numbering. The transmitter buffers all the frames not yet acknowledged positively. Once acknowledged positively the frame is dropped from the buffer.

2.2.7 The P/F bit

The Poll/Final bit enables the command/response mechanism to be carried out . It is recognized only when set to 1 and becomes P in command frames and F in response frames. The response frame to a command frame with P bit set to 1 must carry F bit set to 1. In the normal response mode P/F bit is used for polling secondary stations. The primary station uses the P bit set to 1 to solicit a status response from a secondary station. The secondary station then responds with data or status frame with F bit set to 1.

In the ABM mode, an I-frame sent with P bit set to 1 requires an S-frame response (RR, REJ,RNR) with F bit set to 1, since the I-frame can not be a response. Similarly an RR frame with P bit set to 1 (hence a command) will force an S-frame with F bit set 1 to be sent in the reply. This P/F procedure is called the *check pointing* procedure.

The check pointing procedure has several uses. It is used to force an immediate acknowledgement. On receipt of an I-frame with P bit set to 1, an RR frame with F bit set to 1 will be sent by the receiver immediately, ahead of any I-frame waiting to be transmitted. It can be used to force transmission of an REJ (i.e. a nak) in case of an error, rather than relying on a timeout mechanism. This might, in certain circumstances, speed up error recovery and reduce the number of frames that might have to be transmitted in the event of an error. Finally, the procedure could be used for preparing to take the link down (disconnect). In this case it could be used to clean up the outstanding acks or other control

2.2.8 The I-Frame

The I-frame is used to transfer end-user data between the two stations. This frame may also acknowledge the receipt of data from a transmitting station. In the latter case the information frame in the reverse direction carries the acknowledgement information to the transmitter. This is the well known *piggybacking phenomenon.*

Table 2.1 Four Possible S-Frames and their Functions.

Frame name	S	S	Function
Ready to Receive (RR)	0	0	N(R) acks all frames received up to and including N(R)-1. This is used in the absence of I-frame in the reverse direction (i.e. when no piggybacking is possible
Not ready to Receive (RNR)	1	0	This provides flow control for temporary busy condition. N(R) also acks all frames up to and including N(R)-1.
Reject (REJ)	0	1	N(R) rejects all frames from N(R) on. It positively acks all frames up to and including N(R)-1.
Selective Reject (SREJ)	1	1	Requests for retransmission of a particular frame identified by N(R). Also acks all frames up to and including N(R)-1.

2.2.9 The S-Frame

The S-frame performs control functions such as the acknowledgement of frames and request for temporary suspension of the peer transmission frames (i.e. flow control). Table 2.1 gives the functions of four possible S-frames. The two S bits in the control field decide the type of S-frame being transmitted. The asynchronous balanced mode uses only the first three types of frames [3].

2.2.10 The U-Frame

The U-frame is used for the control purposes only as stated earlier. It is used to perform link initialization, link termination and other control functions. The five M bits in the control field decide the type of U-frame being transmitted. We may, thus, have 32 different U-frames. Further the frames can be commands or responses which means that we can in all have 64 different U-frames. Some of the U-frames are listed in the table 2.2. The function of each frame is briefly discussed below.

1. The *UI format* allows transmission of user data in an unnumbered (unsequenced) frame. The UI-frame is actually a form of connectionless mode link protocol in that the absence of N(S) and N(R) fields precludes flow-controlling and acknowledging frames. The IEEE 802.2 logical link control protocol uses this approach.

2. The *Set Normal Response Mode (SNRM)* places the secondary station in the normal response mode.

3. The *Disconnect (DISC)* places secondary station in the disconnected mode.

4. The *Request Disconnect (RD)* requests for DISC command.

5. The *Unnumbered Poll (UP)* polls stations without regard to sequencing or acknowledgement. Response is optional if poll bit is set to 0.

6. The *Unnumbered Acknowledgement* (UA) response is used by the data link entity to acknowledge the receipt and acceptance of the mode-setting commands (SABME etc.). Received mode setting commands are not processed until the UA response is transmitted. No information field is permitted with UA response. The transmission of a UA response indicates the clearance of any busy condition that was reported by the earlier transmission of an RNR frame by this data link entity.

Table 2.2 Some U-Frames

M	M	M	M	M	Commands	Responses
0	0	0	0	0	UI	UI
0	0	0	0	1	SNRM	
0	0	0	1	0	DISC	RD
0	0	1	0	0	UP	
0	0	1	1	0		UA
0	1	0	0	0	NR0	NR0
0	1	0	0	1	NR1	NR1
0	1	0	1	0	NR2	NR2
0	1	0	1	1	NR3	NR3
1	0	0	0	0	SIM	RIM
1	0	0	0	1		FRMR
1	1	0	0	0	SARM	DM
1	1	0	0	1	RSET	
1	1	0	1	0	SARME	
1	1	0	1	1	SNRME	
1	1	1	0	0	SABM	
1	1	1	0	1	XID	
1	1	1	1	0	SABME	

NR = Not Reserved.

7. The *Set Initialization Mode (SIM)* initializes link control functions in the addressed station.

8. The *Request Initialization Mode* (RIM) format is a request from a secondary station for initialization to a primary station. The secondary can then monitor frames but can only

respond to SIM, DISC, TEST and XID.

9. The *Frame Reject (FRMR)* frame is used to indicate that an improper frame has arrived; one that somehow violates the protocol. One or more of the following conditions have occurred:

   - The receipt of a control field that is undefined (not one of the control field encodings listed in the table 2.2; the table however is incomplete) or not implemented.
   - The receipt of an S or U frame with incorrect length.
   - The receipt of an invalid N(R); the only valid N(R) is in the range from the sequence number of the last acknowledged frame to the sequence number of last transmitted frame
   - The receipt of an I-frame with an information field that exceeds the maximum established length.

   The effect of the FRMR is to abort the connection. Upon receipt of an FRMR, the receiving entity may try to reestablish the connection using the connection establishment procedure.

10. The Set Asynchronous Response Mode (SARM) allows a secondary station to transmit without a poll from the primary station. It places the secondary station in the information transfer state (IS) of ARM.

11. The Disconnect Mode (DM) frame is transmitted from a secondary station to indicate that it is in the disconnect mode.

12. The Reset (RSET) is used as follows: the transmitting station resets its N(S) and receiving station resets its N(R). The command is used for recovery. The previously unacknowledged frames remain unacknowledged.

13. The set Asynchronous Response Mode Extended (SARME) sets SARM with two octets in the control field. This is used for extended sequence numbering and permits the N(S) and N(R) to be seven bits in length.

14. The Set Normal Response Mode Extended (SNRME) sets SNRM with two octets in the control field. This is used for extended sequence numbering and permits the N(S) and N(R) to be seven bits in length.

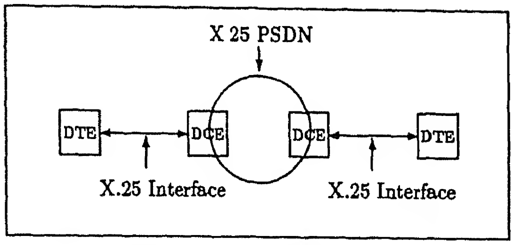


Figure 2.4: X.25 Concept

15. The Set Asynchronous Balanced Mode (SABM) sets mode to ABM in which stations are peers with each other.

16. The exchange identification (XID) is used for two stations to exchange information relating to connection management . When a peer entity receives an XID command it responds with an XID response. XID frame contains an optional field in which ID information is conveyed. No sequence numbers are contained in the control field.

17. The set Asynchronous Balanced Mode Extended (SABME) sets mode to ABM with two octets in the control field for extended sequence numbering.

Thus HDLC is a versatile data link protocol for the data transmission between two systems in a point-to-point or point-to-multipoint link configurations.

2.3 Link Access Protocol - Balanced (LAPB)

LAPB is the data link layer protocol for accessing an X.25 network. It provides for the reliable transfer of a packet from a host or a DTE to an X.25 packet switch, a DCE, which then forwards the packet to its destination. X.25 regulates data flow only between DTE and DCE at each end of the network. The communication details between the DCEs are hidden from the DTEs, the users of the packet switched service. The X.25 concept is depicted in the figure 2.4.

X.25 is organized as a three layer architecture, corresponding to the lowest three layers of the OSI model. Figure 2.5 portrays the three layers of X.25 architecture. The lowest physical layer ensures that a valid physical connection exists between the DTE and the DCE. The CCITT protocol recommendation X.21 is used for this purpose. The X.25 interface architecture is actually

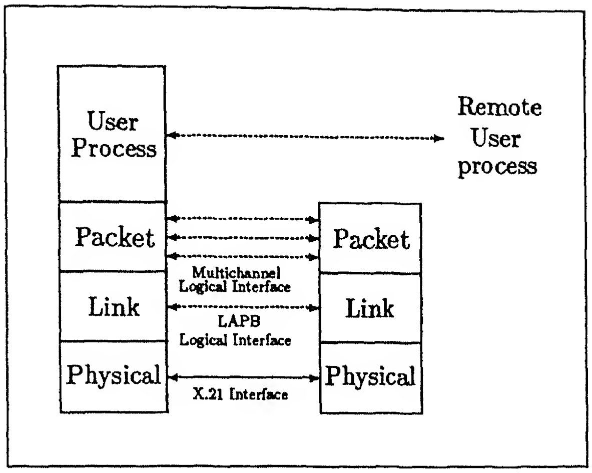


Figure 2.5: X.25 Layers

implemented at the packet level (layer 3). X.25 is based on virtual circuit (VC) connections. Up to 4096 such connections are available between any DTE and the DCE to which it interfaces. A 12 bit address field , defined in the X.25 packet, is used for this purpose . LAPB facilitates multiplexing of the X.25 layer 3 packets on to the single physical link between the DTE and the DCE, giving an *illusion* of multiple paths. LAPB is concerned with providing basic framing and guaranteeing delivery of data from node to node (DCE to DCE). To guarantee data delivery, a rotating window acknowledges each frame received.

This section deals only with the LAPB link level procedures. *LAPB is a subset of the asynchronous balanced mode of HDLC. The frame structure is essentially the HDLC frame structure.* The LAPB link is set up by the user device (DTE) or by the DEC. The SABM/SABME commands set up the link. A station indicates it is able to set up the link by transmitting continuous flags. Prior to link set up, either station can send DISC frame to make certain that all the traffic and modes are cleared. If the link cannot be set up DM frame must be returned.

LAPB has specific procedures for the use of P/F bit. The station upon receiving a SABM/SABME, DISC, S or I frame with the P bit set to 1, must set the F bit to 1 in the next response it transmits. The following conventions apply.

Frame sent with P bit set to 1	**Response required with F bit set to 1**
SABM/SABME, DISC	UA, DM
I (Information Format)	RR, REJ, RNR, FRMR
I (Disconnect Mode)	DM
S (RR,REJ,RNR)	RR, REJ, RNR, FRMR

The address field in the LAPB frame is essential for multidrop lines.~~citeSA:sat~~.

2.4 Link Access Protocol - D channel (LAPD)

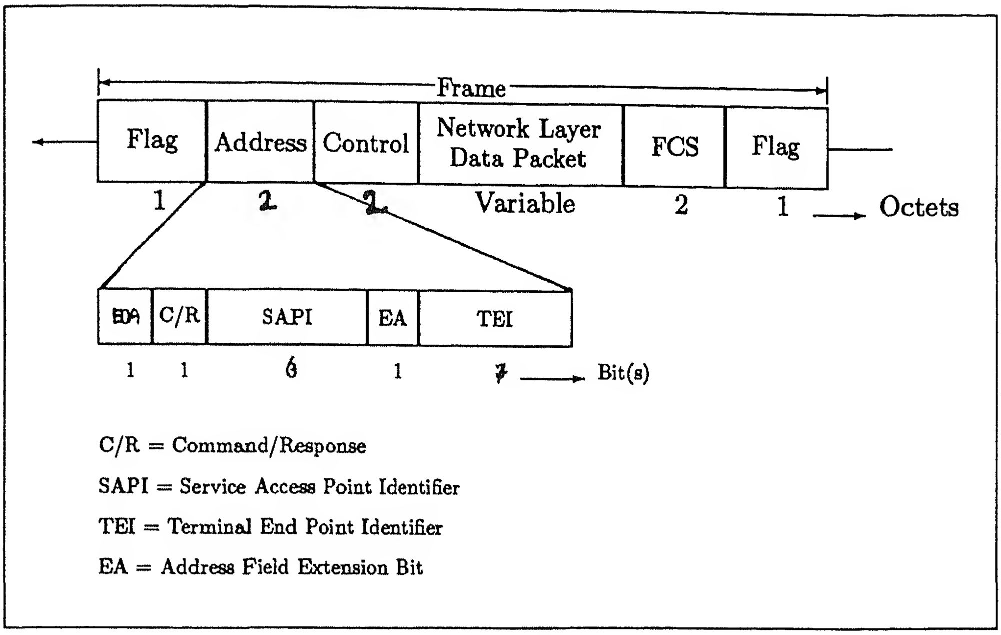
LAPD is a data link control protocol employed to carry the ISDN traffic over D- channel (16 or 64 kbps). It specifies the link access protocol to be used over the D logical channel, which is part of a time multiplexed link between a network subscriber and the ISDN central office [5]. Recommendations Q.921 of CCITT [14] specify the general structure of the LAPD frames, the peer-to-peer procedures for data link layer, the elements for layer-to-layer communication (known as primitives) and the data link control management procedures.

2.4.1 LAPD Frame Structure

Figure 2.6 gives the CCITT standard LAPD frame structure. This frame structure is based on the standard HDLC frame structure. Only the fields which differ from HDLC are discussed below.

Address Field:

The address field format is illustrated in the figure 2.6. The address field in a LAPD frame mandatorily needs two octets unlike HDLC where it is optional. The reason is as follows. LAPD has to deal with two levels of multiplexing. First, at a subscriber site there may be multiple user devices sharing the same physical interface. Second, for each user device, there may be multiple types of traffic: specifically, Packet-switched data and control signalling. To accommodate these levels of multiplexing, LAPD employs a two-part address, consisting of a Terminal End point Identifier (TEI) and a Service Access Point Identifier (SAPI)



C/R = Command/Response

SAPI = Service Access Point Identifier

TEI = Terminal End Point Identifier

EA = Address Field Extension Bit

Figure 2.6: Standard LAPD Frame Structure

Typically, each user device is given a unique TEI. It is possible for a single device to be assigned more than one TEI. This might be the case for a terminal concentrator. TEI assignment occurs either automatically when the equipment first connects to the interface, or manually by the user. The former method has the advantage in the sense that it allows the user to change, add or delete the equipment at will without prior notification to the network administration. In the later method network administration has to maintain a data base for each subscriber that would need to be updated manually. Table 2.3 shows the assignment of TEI numbers.

Table 2.3 TEI Assignments

TEI Value	**TEI User Type**
0 - 63	Non-automatic TEI assignment user equipment
64 - 126	Automatic TEI assignment user equipment
127	Used during automatic TEI assignment

The SAPI identifies a layer 3 user of LAPD and thus corresponds to a layer 3 protocol entity

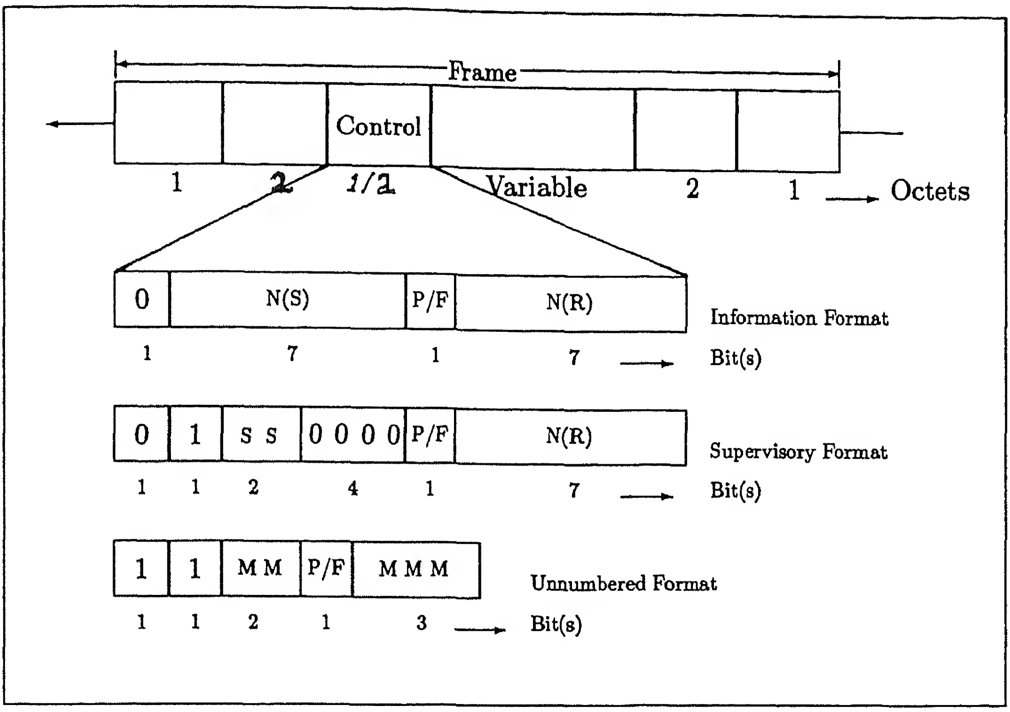


Figure 2.7: LAPD Control Field Structure

within a user device. Four values have been assigned as shown in table 2.4. The SAPI values are unique within a TEI. That is for a given TEI, there is a unique layer 3 entity for a given SAPI. Thus the TEI and the SAPI together uniquely identify a layer 3 entity at a subscriber site. Also these two together uniquely identify a logical connection; in this context, the combination of TEI and SAPI is referred to as Data Link Connection Identifier (DLCI). At any one time LAPD may maintain, multiple logical connections, each with a unique DLCI. Further, at any one time; a layer 3 entity may have only one LAPD logical connection.

Table 2.4 SAPI Assignments

SAPI Value	**SAPI related layer 3 entity or LAPD management entity**
0	Call control procedures
1	Reserved for packet mode communication using I.451 call control procedures
16	Packet communication conforming to X.25 level 3
63	LAPD (layer 2) management procedures
All others	Reserved for future standardization

Thus the SAPI and TEI fields refer to the address of the subscriber layer 3 entity. During transmission the layer 3 entity includes this address in the frame. Frames coming from the network side carry the corresponding destination address and the LAPD entity uses this address to deliver the user data to the appropriate layer 3 entity.

The presence of a 1 in the first bit (EA bit) of an address field octet signals that it is the final octet of the address field. Consequently a two octet address field would have a zero in the EA bit of the first octet and a 1 in the EA bit of the second octet. A single octet address is reserved for the LAPB operation in order to allow a single LAPB data link connection to be multiplexed along with the LAPD data link connections.

Command/Response

The address field includes a Command/Response (C/R) bit which indicates whether a LAPD message is a Command or a Response. User side sends commands with the C/R bit set to 0. It responds with C/R bit set to 1. The network does opposite; it sends commands with C/R bit set to 1 and responses with C/R bit set to 0. Like HDLC the command frames carry the address of the intended receiver and the response frames carry the address of the transmitter. The C/R bit is redundant since the P/F bit included in the control field also indicates whether a frame is a command or a response.

There are two commands and responses in LAPD which are not defined in HDLC structure. These are noted below [12].

Format	**Commands**	**Responses**	**Control Field**
Un-numbered	SI0	SI0	1 1 1 0 P/F 1 1 0
	SI0	SI1	1 1 1 0 P/F 1 1 1

The purpose of the SI0/SI1 commands is to transfer layer 3 information using sequentially acknowledged frames (stop and wait kind of protocol). These command frames contain information fields provided by layer 3. LAPD does not allow information to be placed in SI0/SI1 response frames. These response frames are used during the single frame operation to acknowledge the receipt of SI0 and SI1 commands and to report the loss of frames or any synchronization problen

Control Field:

The control field format is shown in the figure 2.7. This field is two octets long in I and S typ of LAPD frames and only one octet long in the U type LAPD frame. The sequence numbering i thus modulo-128.

Information Field

The information field is present only in I-frames and some unnumbered frames. This field carrie the datagram coming from the network layer. The maximum length of this field, as specified i I.441 for control signals and user data, is 260 octets.

2.4.2 LAPD Services

The LAPD service supports;

- multiple terminals at the user network installation.
- multiple layer-3 entities (X.25 level 3 I.451).

The LAPD standard provides two forms of service to LAPD users:

- the unacknowledged information transfer service

  and

- the acknowledged information transfer service.

The former simply provides for the transfer of frames containing user data with no acknowledge ment. This service does not guarantee that data presented by one user will be delivered to anothe user, nor does it inform the sender if the delivery attempt fails. The service does not provide an flow control or error control mechanism. It supports both Point-to-Point and broadcast kinds. I allows for fast data transfer and is used for management procedure such as alarm messages an messages that need to be broadcast to multiple users.

The acknowledged information transfer service is the more common one, and is similar to th service offered by LAPB and HDLC. With this service a logical connection is established betwee two LAPD users. Three phases occur: connection establishment phase, data transfer phase and connection termination phase. Corresponding to the two types of services offered by LAPD there are two types of operation.

- **Unacknowledged operation:** Layer 3 information is transferred in unnumbered frames. Error detection is used to discard the damaged frames, but there is no error control or flow control.

- **Acknowledged operation:** Layer 3 information is transferred in frames that include sequence number and that are acknowledged. Error control and flow control procedures are included in the protocol.

These two types of operations may co-exist on a single D-channel. With the acknowledged operation it is possible to simultaneously support multiple logical connections. This is analogous to the ability of X.25 level 3 to support multiple virtual connections.

2.5 Serial Line Internet Protocol(SLIP)

SLIP has its origins in the 3COM UNET TCP/IP implementation from the early 1980's. It is merely a packet framing protocol [9]. It provides only the most rudimentary support for sending IP datagrams over asynchronous lines and does not support the datagrams from other protocols and use of synchronous lines. SLIP defines a sequence of characters that frame IP packets on a serial line, and nothing more. It provides no addressing, packet type identification, error detection/correction or compression mechanisms. Because the protocol does so little, it is very easy to implement.

SLIP is commonly used on dedicated serial links and sometimes for dial-up purposes, and is usually used with line speeds between 1200 bps and 19.2 Kbps. It is useful for allowing mixes of hosts and routers communicate with one another (host- host, host-router and router-router are all common SLIP network configurations).

2.5.1 Protocol Frame Structure

Figure 2.8 shows the typical SLIP frame structure. The SLIP protocol defines two special characters: END and ESC. END is hex 0xC0 and ESC is hex 0xDB. To send a packet, coming

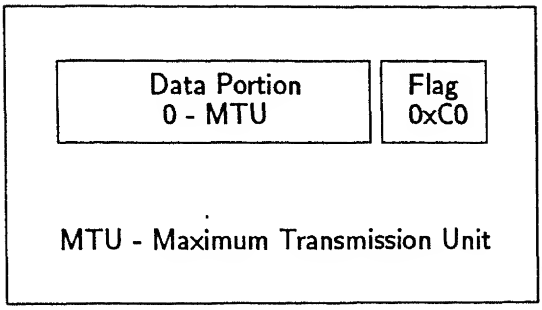


Figure 2.8: Typical SLIP Frame Structure

from the upper layer IP, on to the serial line, a SLIP host simply starts sending the datagram, as it is. When the last byte of datagram is transmitted the host then transmits END character marking the end of datagram. If a data byte is the same code as END character, a two byte sequence of ESC and hex 0xDC is sent instead. If it the same as an ESC character, a two byte sequence of ESC and hex 0xDD is sent instead. Because there is no standard SLIP specification, there is no real defined maximum packet size for SLIP.

2.5.2 Deficiencies

There are several features that many users would like SLIP to provide which it doesn't. In all fairness, SLIP is just a very simple protocol designed quite a long time ago when these problems were not really important issues. The following are commonly perceived shortcomings in the existing SLIP protocol:

Addressing :

Both the systems in a SLIP link need to know each other's IP addresses for routing purposes. Also, when using SLIP for hosts to dial-up a router, the addressing scheme may be quite dynamic and the router may need to inform the dialing host of the host's IP address. SLIP provides no mechanism for hosts to communicate addressing information over a SLIP connection.

Type Identification:

SLIP has no type field. Thus, only one protocol can be run over a SLIP connection. So in a configuration of two DEC systems running, say, both TCP/IP and DECnet, there is no hope of having TCP/IP and DECnet share one serial line between them while using SLIP. While SLIP is *Serial Line IP*, if a serial line connects two multi-protocol systems, those systems should be able to use more than one protocol over the line.

Error Detection/Correction:

Noisy phone lines will corrupt packets in transit. Because the line speed is probably quite low (likely 2400 baud), retransmitting a packet is very expensive. Error detection is not absolutely necessary at the SLIP level because any IP application should detect damaged packets (IP header and UDP and TCP checksums should suffice), although some common applications like NFS usually ignore the checksum and depend on the network media to detect damaged packets. Because it takes so long to retransmit a packet which was corrupted by line noise, it would be efficient if SLIP could provide some sort of simple error correction mechanism of its own.

Compression:

Because dial-up lines are so slow (usually 2400bps), packet compression would cause large improvements in packet throughput. Usually, streams of packets in a single TCP connection have few changed fields in the IP and TCP headers, so a simple compression algorithm might just send the changed parts of the headers instead of the complete headers.

A considerable amount of work has been done by various groups to design and develop a successor to SLIP which will address some or all of these problems. The efforts of Network Working Group of the Internet Engineering Task Force (IETF) in this direction have lead to the development of the Point-to-Point Protocol. This protocol is discussed in detail in the next chapter.

2.6 Summary

The data link layer provides a reliable means to transmit the data across a physical link. Data link control protocols are designed to deal with a variety of physical link characteristics and modes of operation. These protocols can be synchronous or asynchronous. Further they may be bit-oriented, byte-oriented or character-oriented. HDLC is bit-oriented synchronous protocol and is widely used in the networking industry. LAPB is the subset of HDLC and forms the link access protocol for the X.25 packet switching networks. LAPD is another variation of HDLC and specifies the link access protocol for use over the D logical channel of ISDN network. This is part of the time multiplexed link between a network subscriber and an ISDN central office. SLIP is the data link protocol for the transmission of only IP datagrams over the serial line.

Chapter 3

The Point-to-Point Protocol

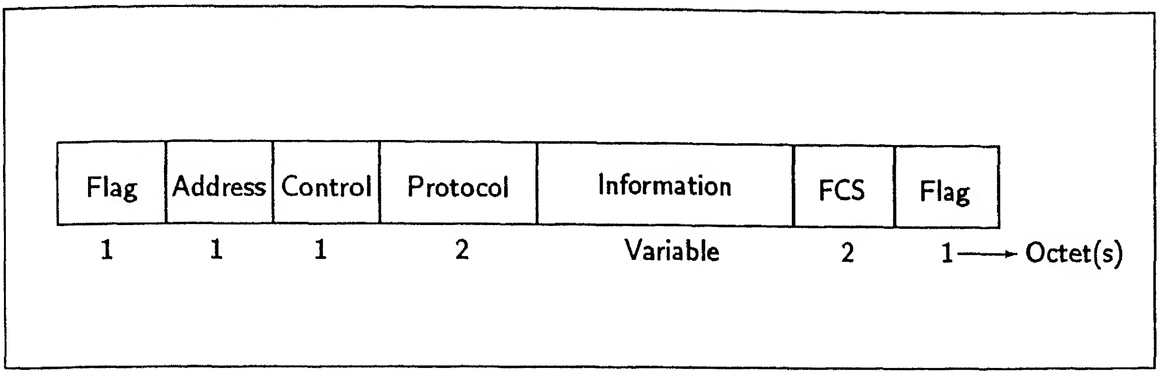
3.1 Introduction

The point-to-point protocol provides a method for transmitting multiprotocol datagrams over the serial point-to-point links. This feature of multiplexing the datagrams from different protocols over the same link is intended to provide a common solution for easy connection of a wide variety of hosts, bridges and routers.

PPP provides an encapsulation protocol over both bit-oriented synchronous links and asynchronous links with 8 bits of data and no parity [7]. These links must be full-duplex, but may be either dedicated or circuit-switched. PPP uses HDLC as a basis for the encapsulation. Further it uses only eight additional octets, by default, to form the basic data link frame. In environments where the bandwidth is at a premium, the additional octets required may be reduced to as few as two octets.

PPP is comprised of the following three main components:

1. A method for encapsulating datagrams over serial links.
2. Link Control Protocol (LCP) for establishing, configuring, and testing the data link connection.



Flag	Address	Control	Protocol	Information	FCS	Flag	
1	1	1	2	Variable	2	1	→ Octet(s)

Figure 3.1: Standard PPP Frame Structure

3. A family of Network Control Protocols (NCPs) for establishing and configuring different network layer protocols.

The basic frame structure of PPP is discussed in the following section. The link control protocol is discussed in the next section. The network control protocol for IP, the IP Control Protocol (IPCP) is discussed in a subsequent section.

3.2 Protocol Frame Structure

The PPP frame format is, basically, derived from the ISO's HDLC with few modifications [15]. Figure 3.1 shows the standard PPP frame structure. As can be seen, except for the 16 bit protocol type field, the PPP frame structure is the same as that of HDLC. However the *address, control* and the *information* fields have the functions which are slightly away from those of HDLC. Further, although, FCS basically does the same function (i.e. error checking) as in HDLC, here it provides various alternatives. These variations are discussed below.

3.2.1 Flags Field

The format and function of this field is similar to that of other standard data link control protocols. Bit stuffing is supported.

Address Field:

Presently the address field is a single octet and contains the binary sequence 11111111 (0xFF), the all stations address. PPP does not assign individual station addresses. Other address lengths and values may be used by prior agreement.

Control Field:

The control field is a single octet and contains the binary sequence 11000000[1] (0xC0), the Unnumbered Information (UI) command with the P/F bit set to zero. Absence of sequence numbers N(S) and N(R) in the control field precludes flow-control and acknowledgement of frames.

Protocol Field:

In order to allow transmission of datagrams belonging to different network layer protocols over the same serial link PPP defines this two octet protocol type field. This is not a field defined by HDLC. Its value identifies the protocol of the datagram/packet encapsulated in the Information field of the PPP data link layer frame. The packets may be from one of the following protocols.

1. Link Control Protocol (LCP)

2. Network Control Protocol (NCP)

3. Any Network Layer Protocol (NLP; IP, IPXCP, OSI and so on)

The most up-to-date values of the protocol field are specified in the most recent *Assigned Numbers RFC* [16].

Frame Check Sequence:

The Frame Check Sequence (FCS) is normally two octets. Other FCS lengths may be used by prior agreement between the two stations involved. The FCS field is calculated over all bits of the *address*, *control*, *protocol* and *information* fields. This does not include FLG sequences or FCS field itself.

The LCP configuration option 9, the FCS-Alternatives provides a method for an implementation to specify an FCS format other than the default one, the CCITT-16 Bits FCS. These formats are:

[1] *This representation of binary numbers is consistent with the ISO and CCITT practice which orders the bits as transmitted (i.e. network bit order). Contrary to this, the standard Internet practice and RFCs follow the MSB to LSB bit ordering.*

- NULL-FCS and
- CCITT 32-bit FCS

This option is negotiated separately in each direction. The negotiated FCS values take effect only during Authentication and Network layer protocol phases. Frames sent during any other phases (LCP, NCPs etc) must contain the default FCS [17].

The link can be subject to loss of state and the LCP can renegotiate at any time. When this occurs, if Alternative-FCS is in effect, the LCP begins renegotiation or termination and sends LCP configure-Request or Terminate-Request packet with the last negotiated FCS value, then changes to the default value and sends a duplicate LCP packet with the default FCS. The Identifier field is not incremented for any such duplicate packet. The peer system, on receipt of this packet changes the default FCS for both transmission and reception. Then, for the packet with duplicate Identifier field a new reply is sent.

The Null-FCS should only be used for those network layer and transport layer protocols which have an end-to-end checksum available, such as TCP/IP or UDP/IP with the checksum enabled. However, it is important to note here that when a configuration (LCP or NCP) or authentication packet is sent, the FCS must be included.

As stated above, for most PPP framed links, the default FCS is CCITT-16 Bits FCS. Some framing techniques and high speed links may use another format as the default FCS. CCITT-32 Bits FCS is normally used in Data Link Control Standards for FDDI and IEEE 802 series [18].

Information Field:

As stated above this contains the datagram of the protocol type specified in the protocol field. Any PPP implementation checks the protocol field to decide the kind of processing the datagram has to undergo. The Information field is one or more octets. The default maximum length is 1500 octets. By negotiation, during the LCP phase, the two stations may decide to use other values for the maximum length.

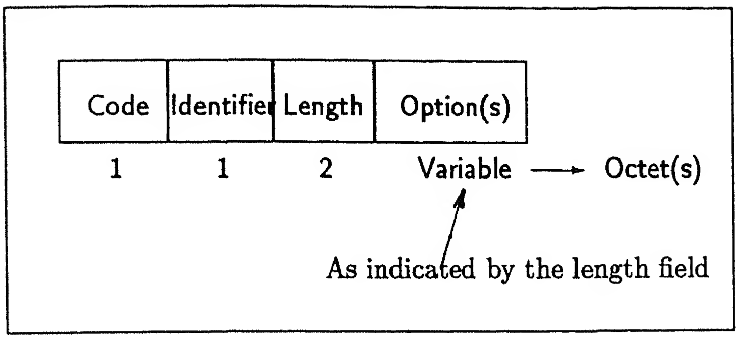


Figure 3.2: Standard LCP Frame Structure

The Information field contains exactly one datagram of any of the aforesaid three protocols; viz. LCP, NCP and NLP.

3.3 Link Control protocol (LCP)

This protocol basically carries out the over all management of the point-to-point link, right from establishing the link with the peer system, configuring it and finally testing the link for data transfer. Exactly one LCP packet is encapsulated in the *information* field of the PPP frame. The LCP packets are classified in to following three categories.

1. Link Configuration Packets used to establish and configure a link.

   (Configure-Request, Configure-Acknowledge, Configure-Nak and Configure-Reject)

2. Link Termination Packets used to terminate a link.

   (Terminate-Request, Terminate-Acknowledge)

3. Link Maintenance Packets used to manage and debug a link.

   (Code-Reject, Protocol-Reject, Echo-Request, Echo-Reply, Discard-Request,Identification and Time-Remaining)

The standard LCP frame structure is shown in figure 3.2.

The *code* field is one octet and identifies the kind of LCP packet. The most up-to-date values of the LCP code field and the corresponding packet type are specified in the most recent *Assigned Numbers RFC* [16]. Current values are assigned as in the table 3.1.

Table 3.1 LCP Packet Types

Code	Packet Type
01	Configure-Request
02	Configure-Ack
03	Configure-Nak
04	Configure-Reject
05	Terminate-Request
06	Terminate-Ack
07	Code-Reject
08	Protocol-Reject
09	Echo-Request
10	Echo-Reply
11	Discard-Request
12	Identification
13	Time-Remaining

The *identifier* field is one octet and aids in matching requests and replies.

The *len* field is two octets and indicates the length of the LCP packet including the Code, Identification, Length and data fields. Octets outside the range of the length field are treated as data link layer padding and are ignored on reception.

The *data* field is zero or more octets as indicated by the length field. The format of data field is determined by the code field. For instance, the data field for the Configure-Request LCP packet (indicated by the code 01 in the code field) carries one or more options to be negotiated.

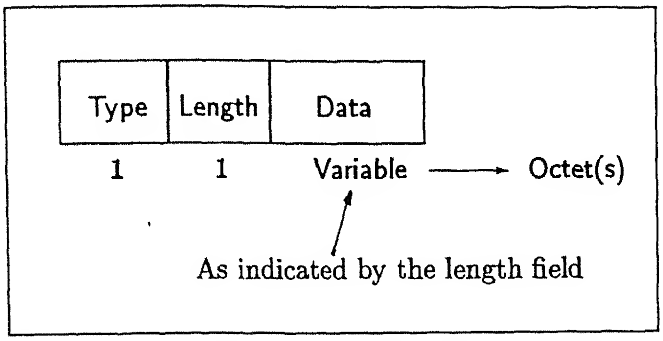


Figure 3.3: LCP Configuration Option Structure

3.3.1 LCP Configuration Options

The LCP configuration option allows modifications to the standard characteristics of a Point-to-Point link to be established. The options to be negotiated are included in the data field of a Configure-Request LCP packet. The end of the list of configuration options is indicated by the length of the LCP packet. If a configuration option is not included in a Configure- Request packet the default value for that option is used. The LCP configuration option format is shown in the figure 3.3.

The *type* field indicates the type of configuration option. The most up-to-date values of the LCP option type are specified in the most recent Assigned Numbers RFC [16]. Current values are assigned as in the table 3.2 below.

The **len** field is one octet and indicates the length of this configuration option including Type, Len and Data fields.

The *data* field is zero or more octets and indicates the value or other information for this configuration option. The format and length of the *data* field is determined by the Type and Len fields. For instance, Type field 9 indicates the configuration option FCS-Alternatives and for this option the *data* field contains one octet indicating which of the three alternatives for FCS is being negotiated. The Compound-Frame option does not carry any *data* field [17].

Table 3.2. LCP Configuration Options

Option Code	Option Type
01	Maximum Receive Unit (MRU)
02	Asynchronous-Control-Character-Map (ACCM)
03	Authentication Protocol (AP)
04	Quality Protocol (QP)
05	Magic Number (MN)
06	Reserved
07	Protocol-Field-Compression (PFC)
08	Address-and-Control-Field Compression (ACFC)
09	FCS Alternatives (FA)
10	Self-Describing-Padding (SDP)
11	Call-Back (CB)
12	Compound Frames (CF)

3.4 Network Control Protocols (NCPs)

Once the link has been established and optional facilities have been negotiated as needed, by the LCP, PPP sends NCP packets to choose and configure one or more network layer protocols. Once each of the chosen network layer protocol has been configured, datagrams from each network layer protocol can be sent over the link. Exactly one NCP packet is encapsulated in the *information* field of PPP data link layer frames where the *protocol* field indicates the protocol number [16] of the corresponding network control protocol. The NCPs use the same packet exchange mechanism as the LCP.

IP-Addresses:

The use of the Configuration Option IP-Addresses has been deprecated. It has been determined through implementation experience that it is difficult to ensure negotiation convergence in all cases using this option [19]. The IP- Address Configuration Option replaces this option, and its use is preferred.

IP-Compression-Protocol:

This Configuration Option provides a way to negotiate the use of a specific compression protocol. Van Jacobson Compressed TCP/IP is one such protocol.

IP-Address:

This Configuration Option provides a way to negotiate the IP address to be used on the local end of the link. It allows the sender of the Configure-Request to state which IP-address is desired, or to request that the peer provide the information. The peer can provide this information by NAKing the option, and returning a valid IP-address.

If negotiation about the remote IP-address is required, and the peer did not provide the option in its Configure-Request, the option should be appended to a Configure-Nak. The value of the IP-address given must be acceptable as the remote IP-address, or indicate a request that the peer provide the information. By default, no IP address is assigned.

The four octet IP-Address contained in the IPCP Configure-Request packet signifies the desired local IP-address of the sender of a Configure-Request. If all four octets are set to zero, it indicates a request that the peer provide the IP-Address information.

3.5 Sending IP Datagrams

Before any IP packets may be communicated, PPP must reach the Network-Layer Protocol phase, and the IP Control Protocol must reach the Opened state. Exactly one IP packet is encapsulated in the Information field of PPP Data Link Layer frames where the Protocol field indicates

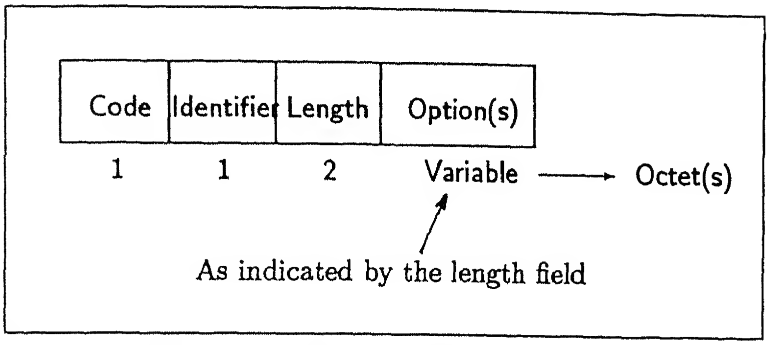


Figure 3.4: Standard IPCP Frame Structure

3.4.1 Internet Protocol Control Protocol (IPCP)

This protocol specifies the network control protocol for IP for sending IP datagrams over the link [19]. IPCP is responsible for configuring, enabling and disabling the IP protocol modules on both ends of the PPP link. IPCP uses the same packet exchange mechanism as the LCP. IPCP packets shall not be exchanged until PPP has successfully completed the LCP phase. The IPCP packets received before this are silently discarded.

Figure 3.4 depicts the standard IPCP frame structure. The IP control protocol is exactly the same as the link control protocol with following exceptions.

Code Field:

IPCP needs to use only Codes 1 through 7 of the LCP codes.
(Configure-Request, Configure-Ack, Configure-Nak, Configure-Reject, Terminate-Request, Terminate-Ack and Code-Reject)
Other Codes are treated as unrecognized and result in Code-Rejects.

Data Link Layer Information Field:

The PPP data link layer frame now carries exactly one IPCP packet in it's information field.

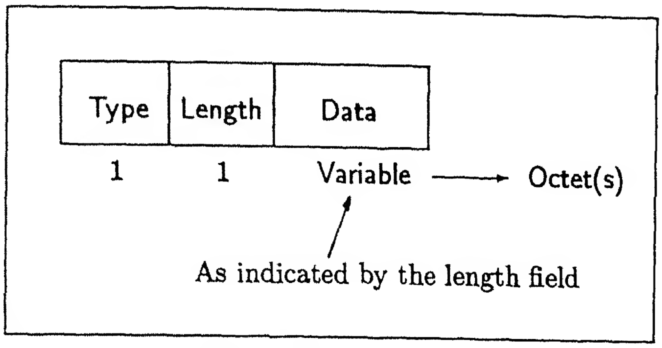


Figure 3.5: IPCP Configuration Option Structure

Data Link Layer Protocol Field:

The PPP Data Link Layer frame's Protocol field now contains the value hex 8021 indicating that the packet being carried in the information field is an IPCP packet.

Configuration Option Types:

IPCP has a distinct set of Configuration Options, which are discussed below. The IPCP configuration option format is shown in the figure 3.5.

3.4.2 IPCP Configuration Options

IPCP Configuration Options allow negotiation of desirable Internet Protocol parameters. IPCP uses the same Configuration Option format defined for LCP, but with a separate set of Options.

The most up-to-date values of the IPCP Option Type field are specified in the most recent *Assigned Numbers* RFC. Current values are assigned as in the table 3.3 below.

Table 3.3. IPCP Configuration Options

Option Code	Option Type
1	IP-Addresses
2	IP-Compression-Protocol
3	IP-Address

type hex 0021 (Internet Protocol). The maximum length of an IP packet transmitted over a PPP link is the same as the maximum length of the Information field of a PPP data link layer frame. Larger IP datagrams must be fragmented as necessary.

3.6 Network Layer Protocols (NLPs)

PPP supports any kind of network layer protocol, unlike SLIP which supports only IP. IP, DECnet, Appletalk, Novell Netware are the well known NLPs supported by PPP. Exactly one datagram, the network layer transmission unit, is encapsulated in the information field of the PPP data link layer frame. Once the PPP completes the Network Control Protocol phase and reaches the opened state in this phase, for the desired network layer protocols, the the datagrams, belonging to the corresponding network layer, can flow over the established serial link. Thus the PPP data link layer permits the multiplexing of datagrams from different network layer protocols.

3.7 Summary

The Serial line protocol for IP, SLIP, provides a nonstandard method for the transmission of IP datagrams over the serial line. Where as PPP offers a standard method for carrying the datagrams, from various network layer protocol modules, including IP, over the same serial line. SLIP just defines a sequence of characters that frame IP packets for transmission on a serial line and nothing more. It does not support addressing , packet type identification , error detection/correction or compression mechanisms. PPP on the other hand involves a Link Control Protocol for the management of the link and a set of network control protocols for the management of the network layer protocols, desiring to use the PPP link. PPP supports all of the features which a SLIP link does support besides several other special features like FCS-Alternatives, Call-Back etc.

Chapter 4

Frame Relay Protocol

4.1 Introduction

With the advent of inexpensive and powerful microprocessors the systems are now capable of processing and storing an enormous amount of data in a short time. As a result it has become necessary not only to increase the data transmission rate between the systems over the network but also the reliability. Further the integration of the services on the same network require the network to transmit voice and video apart from data.

On the other hand the recent advances made in the fiber optic transmission technology is making an enormous amount of bandwidth, in the range of Gbps, available for the reliable, fast and cost effective transmission of digital data. These properties can be used not only to implement new services with high bit rate requirements, but also to simplify the design of the new networks.

The conventional packet switched networks have been designed to cope with a variety of transmission media such as slow, noisy, analog lines through modem connections. As a result the protocol they use on the internode links are complex and hence require a considerable amount of processing at each node which makes high speed implementations difficult. With the fast, highly reliable, digital transmission facilities available, much simpler protocols can be used on the internodal links, thereby reducing the processing time at the nodes. The frame relay technique is based on this principle, wherein the error recovery procedures are removed from the *link level* to the *end user transport level*.

Frame relay, categorized as a *fast packet* technology, combines the statistical multiplexing capability of X.25 with the high speed and low delay characteristics of circuit switching.

4.2 LAN Interconnecting considerations

LANs are deployed to facilitate the exchange of information and sharing of expensive peripherals. Users are today accustomed to the high-speed data transfer capabilities of LANs. Many LAN applications are characterized by *bursty* traffic meaning that they require high band width for small durations. In LANs this is achieved by the high speed statistical multiplexing of the available bandwidth. It is then natural for customers to expect the same performance capabilities from networks providing LAN interconnectivity. These expectations place a significant burden on existing WAN technologies.

As mentioned above the traditional packet-oriented technologies are designed with error checking and recovery procedures necessary to guarantee delivery of packets to intended destinations. Network nodes are responsible for ensuring correctness of packets received prior to forwarding. Each node performs a cyclic redundancy check (CRC) on every packet received and verifies correct sequencing of frames and packets within a defined logical channel.

Protocols such as X.25 were designed to control the transfer of error-free data packets across unreliable transmission facilities. However, these facilities have significantly improved over the past decade. New media such as fiber-optic cable are inherently superior to older media in transmission reliability. Better error characteristics allow error correction to be moved out to the end points of the network. Another factor which has contributed to the decision to move error control out of the network is the availability of intelligent computing systems which are capable of processing the higher layer protocols necessary to provide error recovery procedures on an end-to-end basis.

Frame relay is one of the emerging technologies designed to provide faster and more efficient network services by taking advantage of the latest technical advances. Specifically, three issues motivated the development of frame relay:

1. Spiraling need for additional bandwidth,
2. Increasing power and processing capabilities of computing systems, and
3. Improved transmission facilities.

By making error recovery an end-to-end responsibility, network protocols can be streamlined, removing functions no longer necessary, such as sequence numbers and retransmission procedures. Frame relay eliminates error recovery procedures from network nodes. Frame relay networks perform error checking and discard frames with bad Frame Check Sequences (FCS). Lost frame recovery becomes the responsibility of higher-layer protocols in the end nodes. Frame relay accepts variable-length user information, which means that it operates well with LANs and other synchronous data environments.

4.3 Frame relay versus traditional WAN technologies

To better understand its advantages, frame relay can be compared with some of the major WAN technologies in use today.

4.3.1 Circuit Switching

Circuit-switched data connections, using modems and the public telephone network, are the simplest method of connecting remote locations. Using the Public Switched Telephone Network (PSTN) allows flexible connections anywhere in the world. Once the dial-up connection is established, users pay for connect time rather than actual bandwidth used.

The disadvantages of using circuit-switched connections include:

- The lack of bandwidth for high-speed transmissions.
- Call establishment time, which typically requires many seconds or minutes.

- Connect time charges, which accrue even during idle time.

Therefore, dial-up is not considered an efficient solution for bursty, high-speed applications.

4.3.2 Leased Lines

Leased-line service, using analog or digital facilities, is another option. Leased-line circuits typically provide point-to-point connectivity offering fixed bandwidth capabilities. Dedicated connections limit the flexibility to reconfigure the network without having to add or move physical connections. Moreover, the fixed bandwidth available using dedicated circuits does not offer an economical solution for bursty, bandwidth-intensive traffic.

4.3.3 T1 Networks

Dedicated T1 networks use time-division multiplexing (TDM) to divide the available bandwidth into fixed-length time slots [20]. Each device is assigned one or more time slots for transmission. When a device wishes to transmit, it simply sends the bits in the assigned time slot. Attached devices that send data with relative consistency make the most efficient use of TDM. However, when there is no data to send, the time slot goes unused - resulting in lost bandwidth. T1 networks are best suited to applications with consistent bandwidth requirements that cannot tolerate variable delay, such as voice and video.

As with circuit-switched and leased-line services, T1 is a physical layer solution. There is no protocol processing or translation required. Thus, network latency is reduced to transit delay only. Nonetheless, LAN traffic is not well suited to this rigid approach. Dividing up bandwidth - using a TDM scheme and allowing LAN applications only a small portion of available bandwidth - significantly affects critical response times of bursty LAN applications.

4.3.4 X.25 Packet Switching

X.25 packet switching provides a more efficient method of allocating bandwidth. When a user transmits information, the original message is divided into smaller segments and inserted in a layer 3 packet containing source and destination addressing, as well as additional control information. These packets are routed across the network using the addressing information contained in the layer 3 header.

X.25 uses statistical multiplexing to provide multiple, logical channels over a single physical connection. This method is optimal for bursty applications. However, X.25 provides limited bandwidth. Access services typically range up to 56K bps, nowhere near the multimegabit levels required.

Another problem with X.25 is that its three-layer protocol architecture provides full error and flow control information, requiring large amounts of processing to be done in the network nodes for each data packet received. Protocol processing increases network delay.

4.3.5 Frame Relay

The intent of frame relay is to provide the best of both the circuit-switched and packet-switched worlds. Circuit-switched (and dedicated line) facilities provide fixed bandwidth up to 45M bps and minimum delay, but they lack flexibility. X.25 packet switching uses statistical multiplexing to maximize the use of available bandwidth; however, it requires complex protocol processing and limits access to low-bandwidth facilities.

Frame relay technology combines the statistical multiplexing capability of X.25 with the high speed and low delay characteristics of circuit switching. Currently, frame relay provides data rates up to T1/E1 speeds, and in the near future, a 45M bps service offering will accommodate the growing broadband market. Frame relay has been demonstrated successfully, in the laboratory environments, for speeds up to 100Mbps [21]. In a sense frame relay networks are considered to be the *Next Generation X.25 Networks* [22, 23].

Frame relay performs error detection but does not request retransmission. It eliminates control information such as sequence numbers, acknowledgements, and supervisory frames typically associated with link layer protocols. Stripping out these functions increases throughput because each frame requires significantly less processing. However, frame relay's downside is that flexible frame sizes adversely affect transit delay through the network.

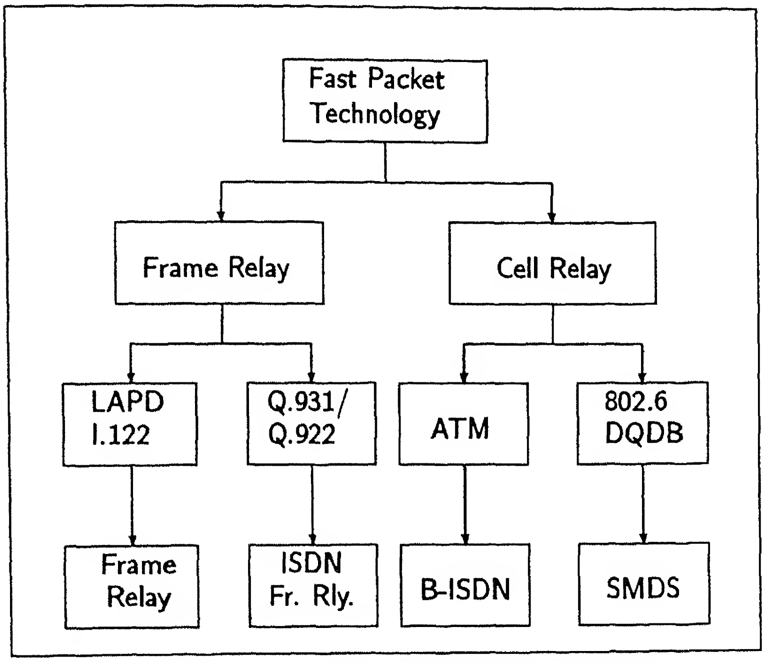


Figure 4.1: Conceptual Broadband Hierarchy

4.4 Broadband Networking

Broadband, in general, defines methodologies for building multimedia networks that provide a common switching fabric for voice, video, and data using reliable, high speed digital communication facilities. This is the basis for the *Information Superhighway* concept which is making a revolutionary impact on the communication industry. The next generation broadband networks will be based on the fast packet switching technology, which combines the best of existing LANs and WANs to transmit voice and video apart from data. Such networks will unite users across the entire spectrum of topologies - local, national, global - at blazing speeds.

One of the most commonly misunderstood aspects of the emerging broadband technology is the relationship of concepts, technologies, standards and the service offerings in which they are packaged. Figure 4.1. shows the conceptual broadband hierarchy and illustrates these key relationships. Although it does not show a comprehensive compendium of all standards and services it does provide an outline of fast-packet networking and shows where the frame relay fits.

The fundamental requirements of all the fast packet implementations are intelligent end systems and low error rate digital transmission facilities which are inherently reliable and can operate at high speeds .

The major difference between the two fast-packet technologies - frame relay and cell relay - are the units of information transferred and the place in the network the protocol is employed. Frame relay transfers information in in variable length *frames* and the cell relay on the other hand transfers information in fixed length *cells*. Frame relay interface is an access arrangement similar to X.25, and is used at the edges or the periphery of the network. Cell relay is used to develop a common, integrated switching fabric for various kinds of information, including frame relay frames, and is used at the core of the network.

4.5 Frame Relay Standards

As a technology, frame relay was initially introduced as an additional packet mode bearer service for ISDN network [25, 24]. Since then extensive work has been carried out in standardizing the frame relay as an independent technical solution. At the same time, ISDN access continues to be a viable entry in to a frame relay network. The underlying principle in the independent frame relay context, however, is very close to that of LAPD.

The functions performed by the LAPD consist of core functions (LAPD Core) and optionable functions (LAPD Upper). The core functions include frame delimiting, virtual-circuit multiplexing, error detection and switching. The optionable functions include error control and flow control. When a network switch performs only the core functions, it is called a frame relay switch. When all the network switches are frame relay switches with only end systems configured with full LAPD functionalities, we have an ISDN frame relay network. Figure 4.2 shows the ISDN frame relay protocol model.

The procedures used in the ISDN frame relay network nodes and end systems are collectively put under the category LAPE, the Link Access Protocol - Extended. Because the address field of the LAPD frame is extended to accommodate the frame relay functionalities [25]. These are

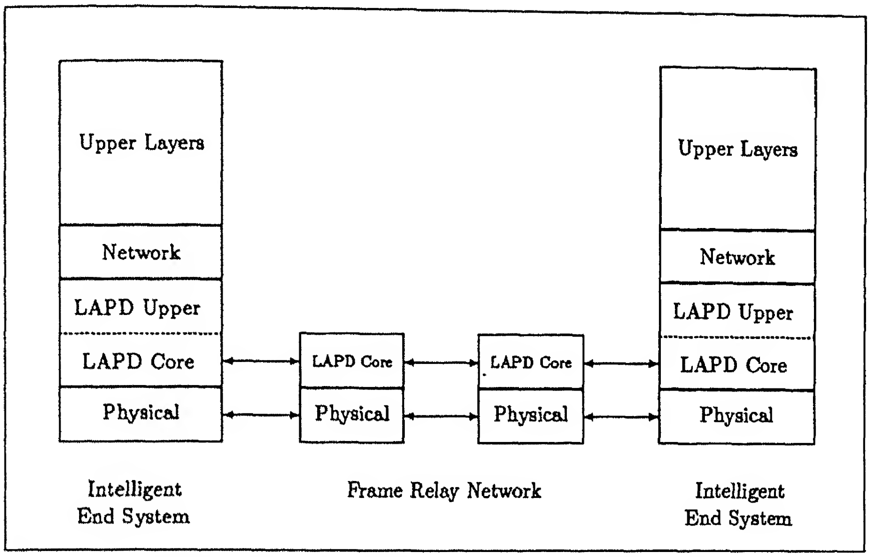


Figure 4.2: The ISDN Frame Relay Protocol Model

specified in the CCITT recommendations Q.922 for Permanent Virtual Circuits (PVCs) and Q.931 for Switched Virtual Circuits (SVCs).

In the independent frame relay context, only the LAPD core functions used with some additional functions for congestion notification. These are collectively referred to as LAPF, Link Access Protocol - Frame Relay. These are specified in the CCITT (now ITU) recommendations I.122 for PVC frame relay services. Procedures for SVC services are still under study [6]. Serious progress will require extensive work because the protocol for setting up virtual circuits is more complex than the other aspects of the frame relay. Until then using PVCs with frame relay (set up via a network management console) offers an opportunity to immediately take advantage of the benefits of frame relay. Figure 4.3 shows the protocol model in the independent frame relay context.

Table 4.1 gives the frame relay related standards as defined by ANSI and CCITT organizations.

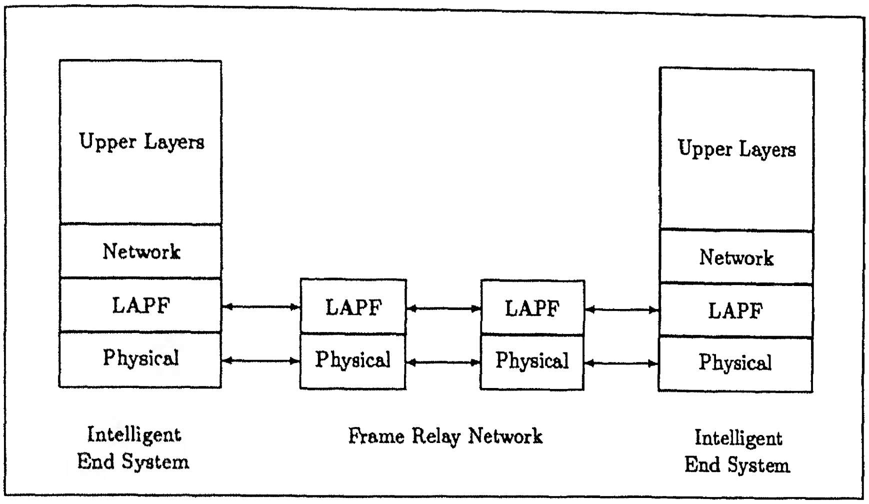


Figure 4.3: The Independent Frame Relay Protocol Model

Table 4.1 Frame relay and related standards

Organization	Standard	Description
ANSI	T1S1/88-2242	Frame Relay Bearer Service - Architectural Framework and Service Description
ANSI	T1.606-1990	Integrated Services Digital Network(ISDN); Frame Relay Bearer Service - Architectural framework and service description
ANSI	T1S1/90-175R4	Addendum to T1.606
ANSI	T1S1/90-214 (T1.6ca)	DSS1 - Core Aspects of Frame Protocol for use with Frame Relay Bearer Service
ANSI	T1S1/90-213 (T1.6fr)	DSS1 - Signalling Specification for Frame Relay Bearer Service
ANSI	T1.606a/1991	Congestion Management - Frame Relay Bearer Service Architectural Framework and Service
ANSI	T1.617f	Description Multiprotocol Encapsulation over Frame Relay
ANSI	T1S1/93-50	FR-SSCS Frame Relay Service Specific Convergence Sublayer
CCITT	I.122/1988	Framework for Providing Additional Packet Mode Bearer Service
CCITT	I.233.1/1991	Frame Relay Bearer Services
CCITT	I.370/1991	Congestion Management in Frame Relay Networks
CCITT	Q.922/1992	ISDN Data Link layer Specifications for Frame Mode Bearer Service
CCITT	Q.931	ISDN Network Protocol (Proposed SVC approach for frame relay)
CCITT	Q.933/1992	DSS1 Signalling Specifications for Frame Mode Basic Call Control

4.6 A Typical LAN Interconnection Scenario

An important characteristic of traditional LAN interconnection technology is a permanent physical connection between any two sites. As the figure 4.4 shows, normally bridges or routers from respective LANs are interconnected using serial communication cards or serial communication ports.

This method of connecting many LANs with routers to make one large Wide Area Network (WAN) is simple and easy when small number of LANs are involved. However, when it is desirable to connect many LANs in this fashion, performance drops off sharply, costs climb exponentially and the collection of interconnected equipment is almost unmanageable. For, as the figure 4.4 shows, a

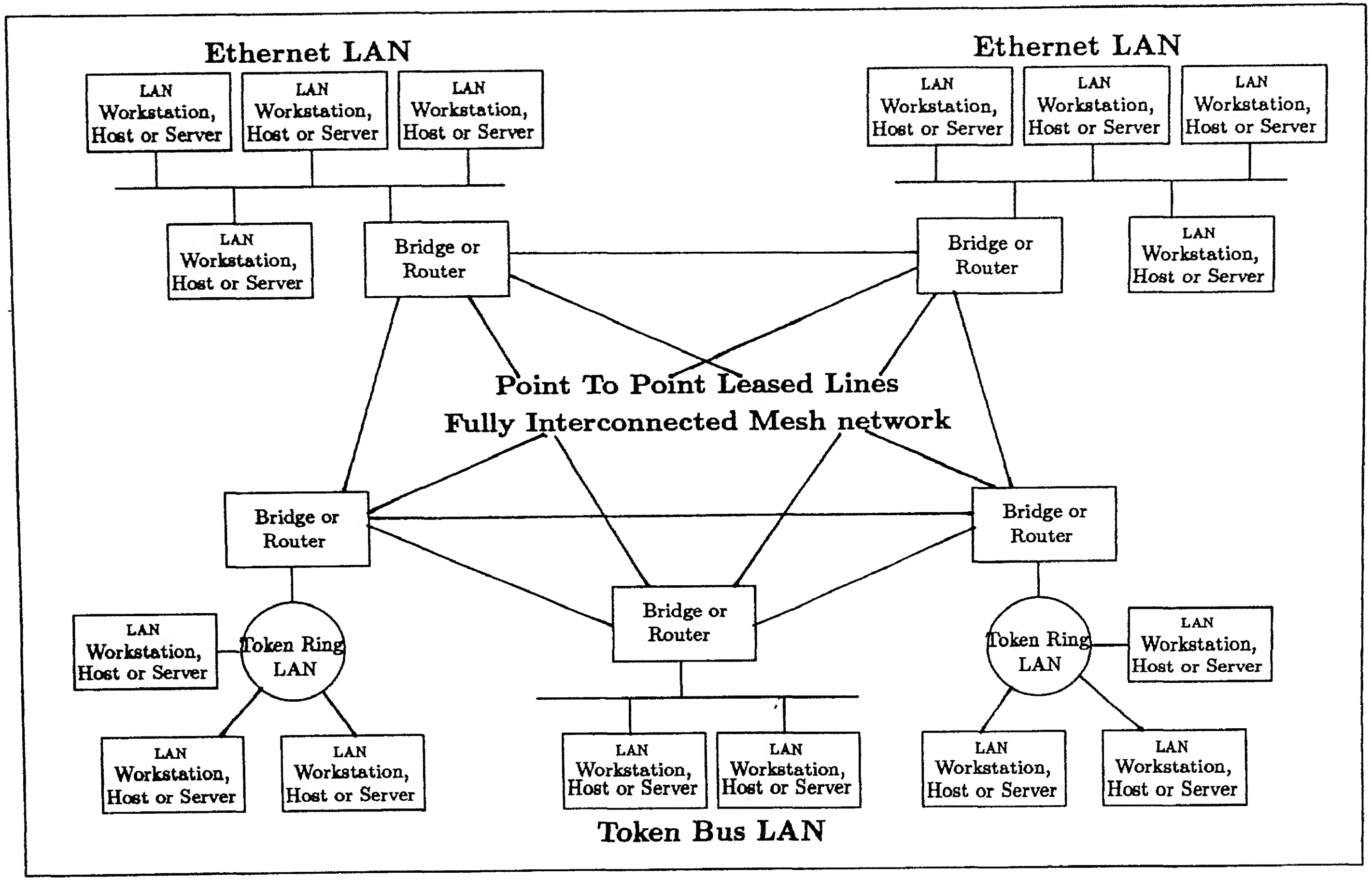

Ethernet LAN
LAN Workstation, Host or Server
LAN Workstation, Host or Server
LAN Workstation, Host or Server
LAN Workstation, Host or Server
Bridge or Router
Ethernet LAN
LAN Workstation, Host or Server
LAN Workstation, Host or Server
LAN Workstation, Host or Server
Bridge or Router
LAN Workstation, Host or Server
Point To Point Leased Lines
Fully Interconnected Mesh network
Bridge or Router
Bridge or Router
Bridge or Router
LAN Workstation, Host or Server
Token Ring LAN
Token Ring LAN
LAN Workstation, Host or Server
LAN Workstation, Host or Server
LAN Workstation, Host or Server
LAN Workstation, Host or Server
LAN Workstation, Host or Server
LAN Workstation, Host or Server
LAN Workstation, Host or Server
Token Bus LAN

Figure 4.4: LAN Interconnection without Frame Relay

full mesh interconnection is required to guarantee low latencies - by reducing the number of hops to minimum - and to assure alternate routes in case of failures. And the number of paths required for a full-mesh interconnected network rises at the rate of n(n-1)/2, where 'n' is the number of devices being interconnected. The cost model deteriorates, the performance falls drastically and the manageability becomes more complex with more than just five or six remote LANs.

The alternative to a physical mesh internetwork is a logical mesh internetwork as shown in the figure 4.5 for a frame relay network. LAN interconnection with frame relay is achieved by providing a single multiplexed physical connection for each LAN device - Bridge, Router, Customer Premises Equipment (CEP) - to a very high speed backbone. Each physical connection carries many logical connections that can be used to create a logical mesh internetwork. Thus *a frame relay network acts like a LAN in itself* rather than just a point to point link,transferring bits between the fixed locations. A sending device simply has to place an addressed frame in to the network and it is transported quickly to it's destination. Utilizing the single physical Frame Relay Interface (FRI) a LAN device such as router or any other frame relay equipped device, may send data directly to any other device in the network by specifying the destination's address.

The high speed backbone network to which a FRI is connected must provide bandwidth on demand in order for a user of frame relay to realize all of the benefits of frame relay. The capability of a network to absorb data as the user wants to transmit is called band-width-on-demand. If the sender has a large burst of data to send - a very characteristic of LANs - the bandwidth is made available and the data is moved in to the network for transmission. If little or no data is to be transmitted, little or no bandwidth is provided. Since users are assigned independent logical paths over a single physical connection it is possible to provide all the bandwidth to a single user when others are inactive. However, as other users go on becoming active the band width assigned to an individual user is accordingly reduced down to his permitted bandwidth, as defined by the CIR (Committed Information Rate). The CIR defines the upper bound on the data that can be transmitted by an individual in the presence of *all* the other users of the interface.

The frame relay design approach illustrated in the figure 4.5 reduces total recurring costs,

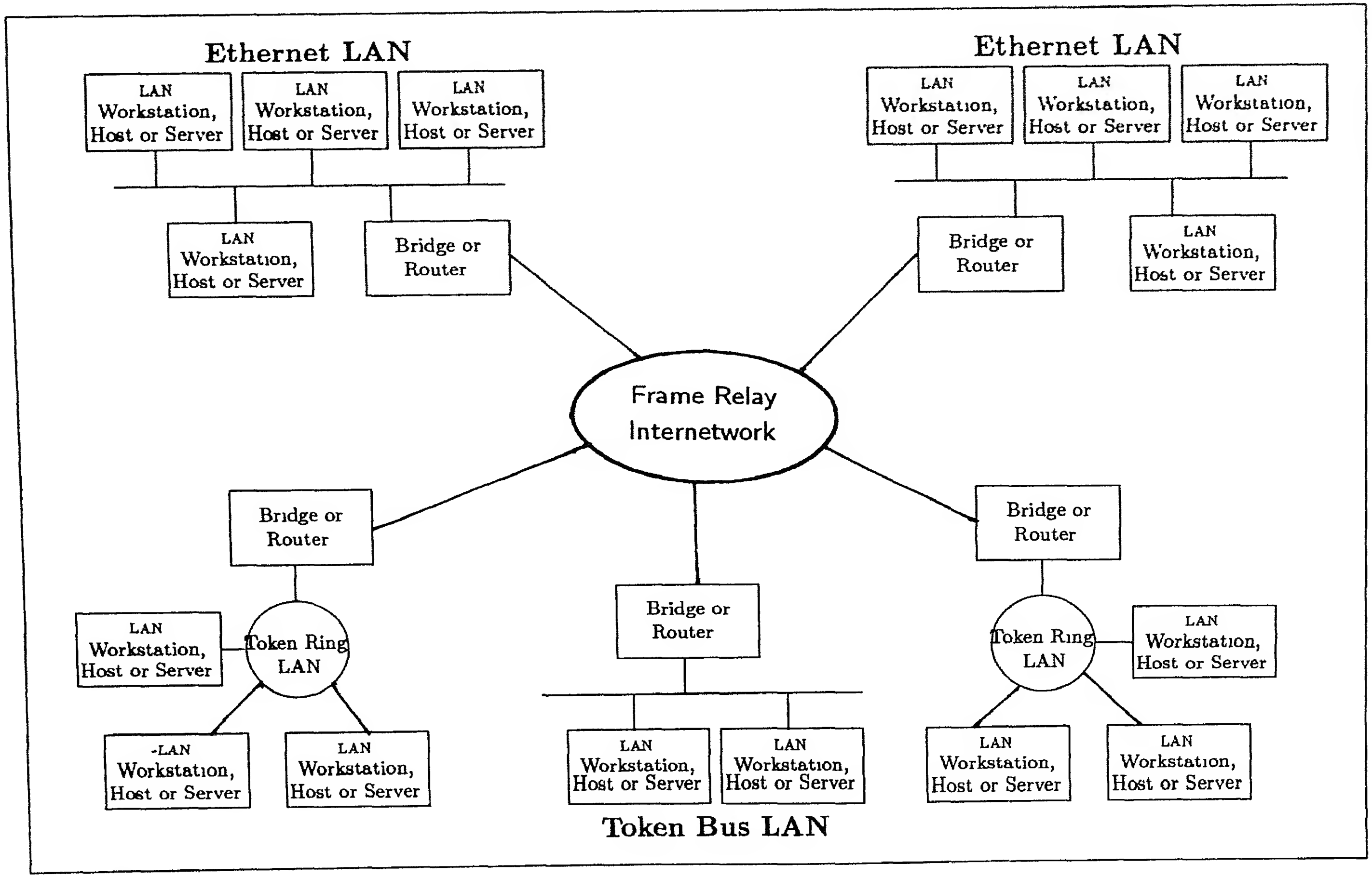

Ethernet LAN
LAN Workstation, Host or Server
LAN Workstation, Host or Server
LAN Workstation, Host or Server
LAN Workstation, Host or Server
Bridge or Router
Ethernet LAN
LAN Workstation, Host or Server
LAN Workstation, Host or Server
LAN Workstation, Host or Server
Bridge or Router
LAN Workstation, Host or Server
Frame Relay Internetwork
Bridge or Router
LAN Workstation, Host or Server
Token Ring LAN
-LAN Workstation, Host or Server
LAN Workstation, Host or Server
Bridge or Router
LAN Workstation, Host or Server
LAN Workstation, Host or Server
Token Bus LAN
Bridge or Router
Token Ring LAN
LAN Workstation, Host or Server
LAN Workstation, Host or Server
LAN Workstation, Host or Server

Figure 4.5: LAN Interconnection with Frame Relay

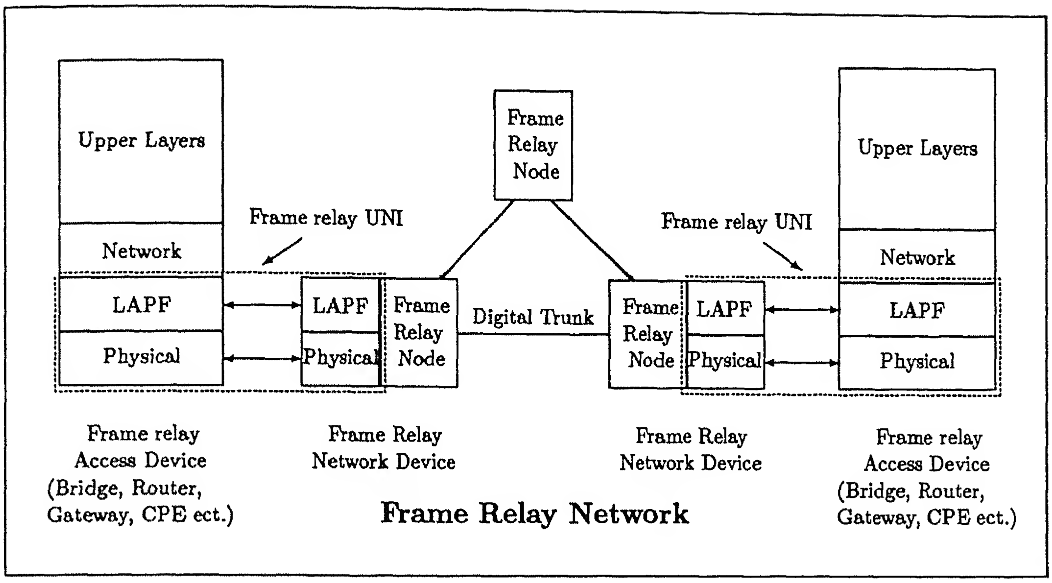


Figure 4.6: Frame Relay Protocol Model

reduces network induced latency, and increases reliability. Recurring costs are reduced because a single higher capacity connection to a common network backbone carries all the traffic, regardless of destination (which is not true for the traditional LAN interconnection approach depicted in the figure 4.4). When used with an efficient, high speed, bandwidth-on-demand frame relay backbone the network induced latency is reduced because hop counts decrease, processing time at nodes decrease and hence the available bandwidth increases. Reliability can also be increased because the backbone network supporting frame relays use highspeed low bit error rate digital transmission facilities.

4.7 The Protocol Model

Figure 4.6 depicts the frame relay protocol model for the User-to-Network-Interface (UNI). Like X.25, the frame relay UNI is asymmetrical, meaning that the devices on the two sides of the interface have different roles to play. The intelligent end systems - routers, bridges, CPEs etc. - are defined as frame relay access devices (FRADs). The intelligent nodal processors at the other end of the UNI are defined as frame relay network devices (FRNDs) [6]. An OSI level 2 data transfer protocol operates between FRNDs and transports data from one end of the network to the other.

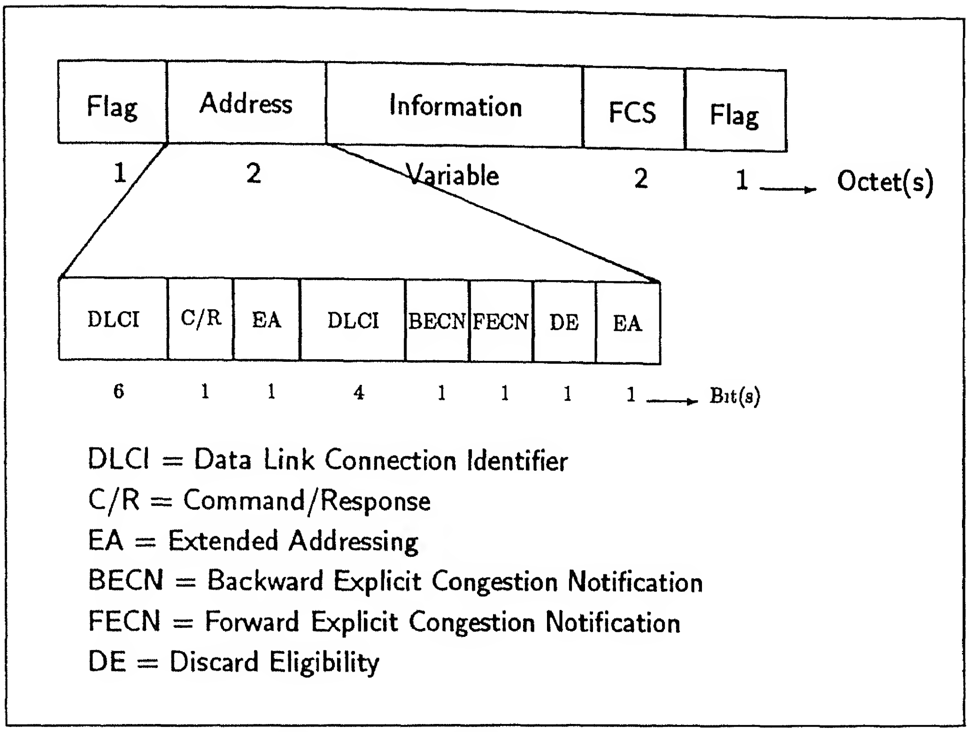


Figure 4.7: LAPF Frame Structure

A second layer 2 protocol, the control protocol, operates between the FRAD and the FRND to which it is connected. Thus frame relay is a packet-mode service similar to X.25; data is enveloped in individually addressed packets that are statistically multiplexed over a single physical facility. But unlike X.25, which operates up through OSI layer 3, frame relay uses only OSI layers 1 and 2 for data transfer. Routing information commonly found in the layer 3 (network layer) header [26] is moved into the layer 2 (data link layer) header, reducing the amount of protocol processing required at each node in a frame relay network [27].

4.8 The Protocol

The LAPF frame format as specified by CCITT Q.922/I.122 is shown in the figure 4.7. It may be seen that the control field, which is a common feature in any conventional communication protocol, is absent in LAPF. This is because the error recovery function is moved out of the network which in turn obviates the fields such as sequence numbers and other control elements normally found in the control field [chapter 2]. Hence LAPF requires less overhead in each frame than the conventional bit-oriented protocols. The functions of the fields in a LAPF frame are briefly explained below.

Flags:

As in other bit-oriented protocols, the flags provide frame delimiting and synchronization. In order to assure that no bit patterns in the payload portion of the frame inadvertently match the flag pattern of 01111110, frame relay performs *zero bit insertion.*

Address field:

Provides routing and control information. The functions of the bits specified in the address field are discussed below.

1. **DLCI:** This actually is the addressing mechanism of frame relay. It identifies the virtual connection (presently only PVC implementations are in use) corresponding to a particular destination. Some DLCIs are reserved for signaling, management and maintenance operations. For instance, DLCI0 is reserved for in-channel call control signaling as defined by CCITT Q.931/ANSI T1.6fr standards. These standards are, however, well suited for SVC frame relay service and do not apply to present implementations of frame relay. DLCI 1023 is reserved for Local Management Interface (LMI) communication. This is used for sending link layer control messages from the network to the user device. DLCIs 1 through 15 and 1008 - 1022 are reserved for future use. This leaves the 922 DLCIs, from 16 through and including 1007 available for frame relay PVCs.

   DLCIs have local significance only. A frame relay connection that goes from node A, in the figure 4.8, to node B may have a different DLCI on the UNI at each end of the connection. For instance, node A may use DLCI 18 to identify node B. Node B may use DLCI 27 to identify node A and DLCI 18 to identify node C. Similarly node A may use DLCI 27 to identify C. Also it is possible that node C uses DLCI 18 to identify node B.

2. **The C/R Bit:** The Command/Response bit identifies a frame as a command or response. This bit, however, is not used by the frame relay data link protocol and may be set to any value by the frame relay access device. This bit is carried transparently by the frame relay network.

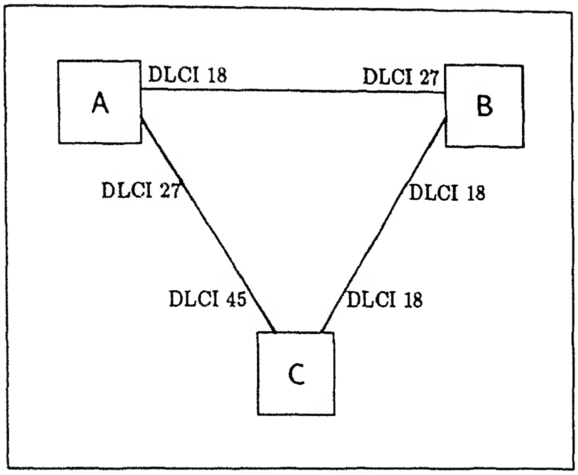


Figure 4.8: LAPF DLCI Assignment

3. **The EA Bit:** The Extended Address bit indicates whether extended addressing is used or not. If extended addressing is used the address field will be 3 or 4 octets, as against the normal 2 octets. An EA bit of 0 indicates that another address octet follows follows this one, and 1 indicates that this is the last octet of address field. Thus in the case of two octet address field, the first octet will have this bit set to 0 and in the second octet this bit will be set to 1.

4. **The FECN/BECN bits:** These bits provide a method for network nodes to identify congestion within the network to destination and source end systems.

5. The DE Bit: This bit is used by a network node to identify frames eligible for discard in preference to other frames if network encounters congestion. The frame relay network would discard frames with a DE bit set to 1 instead of frames with DE bit set to 0 when the network becomes congested and no longer carry all the data presented to it by the user.

The Infirmation Field:

This field contains user data from attached devices for transport over the frame relay network. The default maximum value is 260 octets. However, frame relay networks support a negotiated maximum value of 1600 octets for interconnecting LANs in order to minimize the need for segmentation and reassembly.

The FCS Field:

This field is used to verify the integrity of the frame. It is important to note that FCS is calculated by the source frame relay device and recalculated by the destination frame relay device. If the two FCSs do not match the frame is discarded. It may be recalled that there is no method to request retransmission within frame relay. Retransmissions, if needed, will be handled by a higher layer protocols, in the end systems.

4.9 Congestion Management

Connecting high-speed networks over the wide area presents network designers with a new set of challenges. As in any packet switching network, one of the key areas in the design of a frame relay network is congestion control. At any network node, if the rate at which frames arrive and queue up exceeds the rate at which the node can process and transmit, the queue size grows without bounds and the delay experienced by the frame goes to infinity, thereby leading to severe congestion.

Figure 4.9 shows the effect of congestion in general terms[28]. It may be noted there that, at light loads, the network throughput increases as the load increases. As the load continues to increase, a point is reached (point A in figure) beyond which the throughput of the network increases at a rate slower than that of the offered load. This is due to network entry in to a mild-congestion state. At this level, the network continues to cope with the load, although with increased delays.

As the load on the network further increases, a point is eventually reached (point B in figure) beyond which throughput drops with increased offered load. The reason for this is that the buffers at each node are of finite size. When the buffers at any node become full the node starts discarding frames. As a result, the sources must retransmit the discarded frames in addition to new frames. As more and more frames are retransmitted, the load on the system grows and more buffers become saturated. Even successfully delivered frames may be retransmitted because the acknowledgement from a higher layer arrived too late. Avoiding these catastrophic events is the task of congestion control.

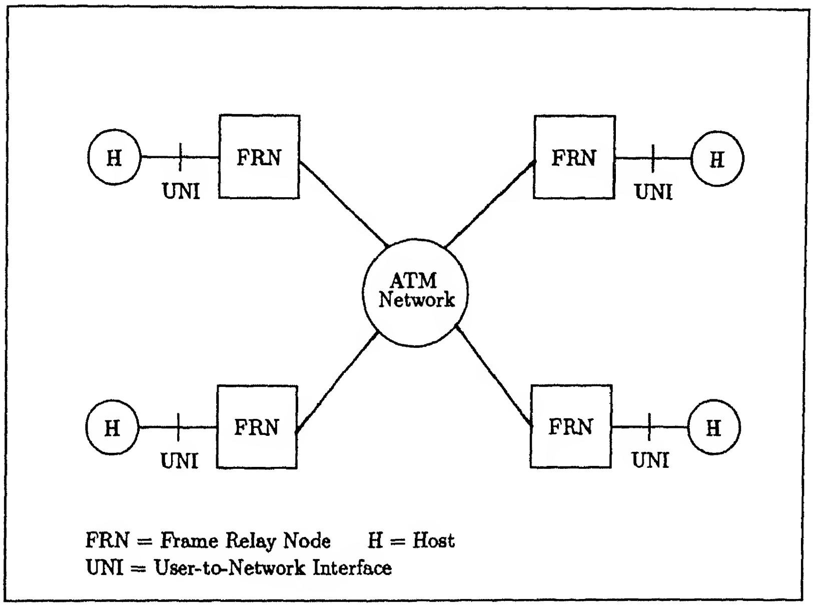

H
FRN
UNI
ATM Network
FRN = Frame Relay Node H = Host
UNI = User-to-Network Interface

Figure 4.10: Interconnection of Frame Relay through ATM

While considering the possible relationship between frame relay and ATM networks, two different scenarios can be envisaged.

1. The interconnection of frame relay through ATM

   and

2. The interworking between frame relay and ATM networks.

In the interconnection scenario, shown in the figure 4.10 the ATM network provides the transmission infrastructure for a higher layer frame relay network. All the users are directly connected to the frame relay nodes, which implement also the interworking functionality with the ATM networks.

In the interworking scenario, shown in the figure 4.11, communication can be established between two generic users, regardless of the network to which they are connected. Inside the ATM terminals or Inter Working Units (IWUs) - bridges, routers, gateways - the LAPF protocol is replaced by a Frame Relay Service Specific Convergence Sublayer (FR-SSCS) of the ATM Adaption layer[1] type 5 (AAL5) [33]. Reference [34] gives a good description of the traffic management aspects ATM local area networks.

[1] Adaption protocols above the ATM layer have been defined in international standardization committees in order to map service specific protocols to the service-independent ATM protocol.

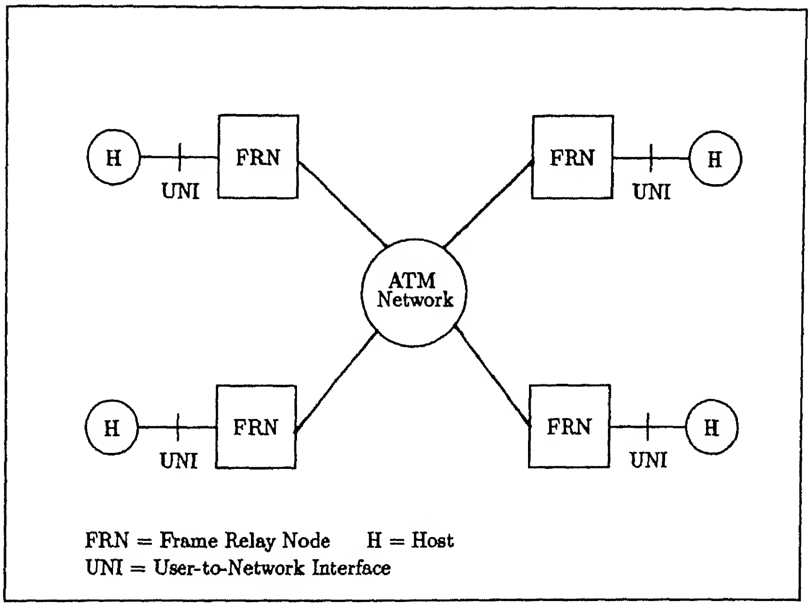

H
UNI
FRN
FRN
UNI
H
ATM Network
H
UNI
FRN
FRN
UNI
H
FRN = Frame Relay Node H = Host
UNI = User-to-Network Interface

Figure 4.10: Interconnection of Frame Relay through ATM

While considering the possible relationship between frame relay and ATM networks, two different scenarios can be envisaged.

1. The interconnection of frame relay through ATM

   and

2. The interworking between frame relay and ATM networks.

In the interconnection scenario, shown in the figure 4.10 the ATM network provides the transmission infrastructure for a higher layer frame relay network. All the users are directly connected to the frame relay nodes, which implement also the interworking functionality with the ATM networks.

In the interworking scenario, shown in the figure 4.11, communication can be established between two generic users, regardless of the network to which they are connected. Inside the ATM terminals or Inter Working Units (IWUs) - bridges, routers, gateways - the LAPF protocol is replaced by a Frame Relay Service Specific Convergence Sublayer (FR-SSCS) of the ATM Adaption layer[1] type 5 (AAL5) [33]. Reference [34] gives a good description of the traffic management aspects ATM local area networks.

[1] Adaption protocols above the ATM layer have been defined in international standardization committees in order to map service specific protocols to the service-independent ATM protocol.

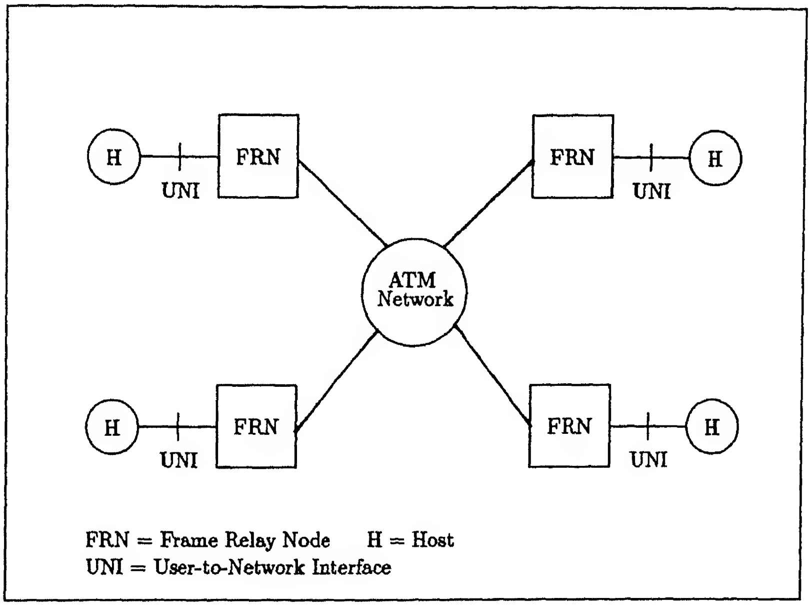

H
UNI
FRN
FRN
UNI
H
ATM
Network
H
UNI
FRN
FRN
UNI
H
FRN = Frame Relay Node H = Host
UNI = User-to-Network Interface

Figure 4.10: Interconnection of Frame Relay through ATM

While considering the possible relationship between frame relay and ATM networks, two different scenarios can be envisaged.

1. The interconnection of frame relay through ATM

   and

2. The interworking between frame relay and ATM networks.

In the interconnection scenario, shown in the figure 4.10 the ATM network provides the transmission infrastructure for a higher layer frame relay network. All the users are directly connected to the frame relay nodes, which implement also the interworking functionality with the ATM networks.

In the interworking scenario, shown in the figure 4.11, communication can be established between two generic users, regardless of the network to which they are connected. Inside the ATM terminals or Inter Working Units (IWUs) - bridges, routers, gateways - the LAPF protocol is replaced by a Frame Relay Service Specific Convergence Sublayer (FR-SSCS) of the ATM Adaption layer[1] type 5 (AAL5) [33]. Reference [34] gives a good description of the traffic management aspects ATM local area networks.

[1] Adaption protocols above the ATM layer have been defined in international standardization committees in order to map service specific protocols to the service-independent ATM protocol.

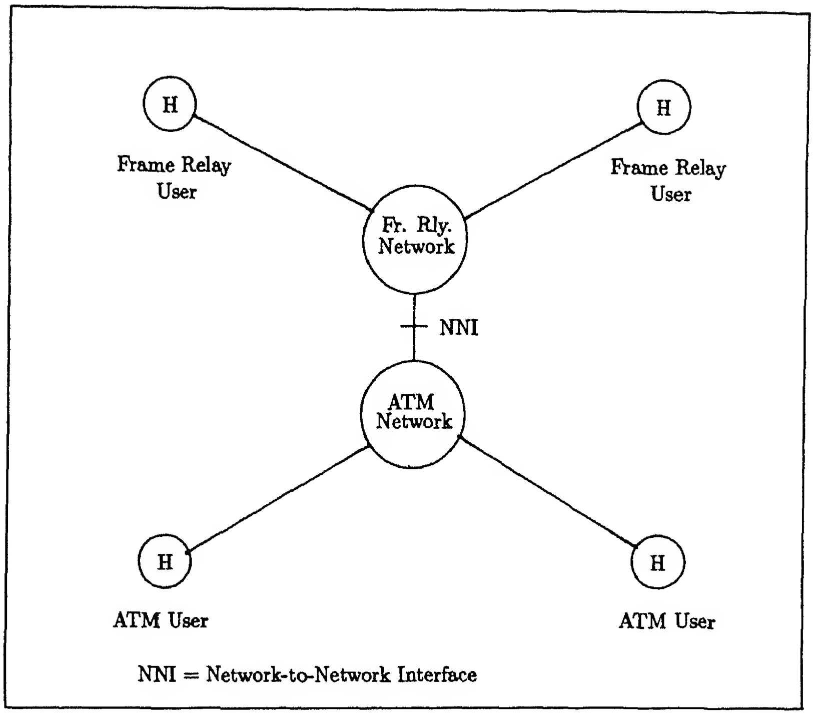


Figure 4.11: Interworking between Frame Relay and ATM Networks

4.11 Frame Relay Forum

Frame relay has emerged in the networking industry in a variety of ways, including the formation of interest groups and the establishment of standards and implementation strategies.

Founded (by Cisco,DEC,StrataCom,and Northern Telecom) in 1991, the Frame Relay Forum (FRF) is a consortium of manufacturers, service providers, and end users of frame relay, working together to advance frame relay as a technology platform. The FRF has been instrumental in developing implementation agreements for vendors of frame relay products and services. The FRF has been a driving force behind the development of intra- and inter-carrier implementation, interoperability, and service agreements related to frame relay.

4.12 Multiprotocol Support

Frame relay networks provide interconnection of LANs using bridges and routers. The Internet Engineering Task Force-Request For Comment (IETF-RFC) 1490, Multiprotocol Interconnect Over

Frame Relays, describes the use of frame relay networks as a backbone for interconnecting other network technologies [35].

All protocols transported over a frame relay backbone are encapsulated within a LAP-F frame using CCITT Q.922 Annex A. The Protocol Data Unit (PDU) received from the attached network and additional control information necessary to identify the protocol being transported are carried in the information field of LAP-F.

A Network Layer Protocol ID (NLPID) identifies the transported protocol so that the receiver can pass the incoming PDU to the appropriate higher-layer protocol. Administered by the ISO and CCITT, NLPIDs are currently defined for many different protocols, including Internet Protocol (IP), Connectionless Network Layer Protocol (CLNP), and IEEE Subnetwork Access Protocol (SNAP).

4.13 Summary

Frame relay is a standardized interface that provides multiplexed access to bandwidth-on-demand backbone networks and provides LAN like performance over a wide area. It is designed to support the high-speed transmission requirements necessary for LAN-to-LAN interconnection and other bandwidth-intensive applications. Frame relay has burst onto the networking scene, with virtually every carrier offering some level of service. Support for multiple logical connections across a single physical facility encourages increased connectivity without requiring additional physical hardware and interfaces. Reduced processing time at each frame relay node reduces transit delay and response time critical to end user applications. Essentially, this is the reduction of number of seconds required to move a specified number of bytes. In other words, **instead of reducing the number of bits per second frame relay reduces the number of seconds per bit.** All these factors lead consumers to frame relay that provides improved performance at reduced costs.

Chapter 5

Design of PPP-IP Interface

In chapter 1 it is pointed out that the point-to-point protocol implemented in the ERNET LAB is able to receive data packets in the polled fashion only. Further the data exchange between the two systems is possible only at the data link layer. The PPP is not able to exchange data packets with the IP network layer. In a router the routing decisions are taken at the network layer. Therefore in order to enable the router to route the data packets on the serial line, this PPP module is required to be interfaced with the IP module.

In this chapter an overview of the IITK router software is presented first. This is followed by the packet forwarding process of the router using the Ethernet LAN Network Interface Adapters. Two possible approaches for the packet transmission by the router on serial links are also presented. This helps in understanding the enhancements to be made in the PPP module for interfacing it with the IP module. The PPP-IP interface design considerations are discussed next. The chapter concludes with the presentation of the enhancements made in the PPP module.

5.1 IITK Router Software:An overview

A router basically operates at the network layer, layer 3 of the 7 layer OSI protocol model. The data link layer (layer 2) and the physical layer (layer 1) are also involved in the operation of a router. A brief description of these layers as implemented in the IITK routers is presented below.

5.1.1 Network Layer

The router code being used in ERNET LAB consists of a complete implementation of Internet Protocol (IP) and User Datagram Protocol (UDP) at the network layer. It also has necessary device driver code for some standard network devices like Ethernet and Token Ring and non-network devices like programmable timer and keyboard. But, there is no such device driver provided for the serial link.

The code is independent of the CPU in the sense that it can operate well with the IBM-AT, MICRO VAX and M68K kind of systems. Only the corresponding software switches are to be set on. Since it operates in the DOS environment, which does not support multitasking, it is not possible to have multiple processes running concurrently. However the IP router code provides a mechanism for scheduling several processes (timer process, keyboard polling process and as many input processes as there are NIAs on the system for processing incoming datagrams) required for the router operation and creates an illusion of multitasking. A *Process Scheduler* accomplishes this by switching the CPU among several processes in a predetermined and dynamic manner. This is essentially the allocation of the *CPU time*, one of the vital system resources, for different processes. The software module *pswitch()* in the IP router code carries out this switching and scheduling task.

Memory is another important system resource. A router system must very judiciously allocate it's available memory for carrying out various tasks in addition to the basic function of datagram reception, storage and forwarding. The IP router code has it's own memory management scheme. The software module *lsize()* in the router code does the job of allocating the memory resource to several processes running on the router. It also allocates the necessary memory for the reception and transmission of the datagrams.

Thus the two vital system resources the *CPU time* and the *memory* are dynamically shared by several processes.

5.1.2 Data Link Layer

The data link layer basically provides the reliable data transfer support for the error prone and hence the unreliable physical medium. In the ERNET Lab the router system uses the PCLINK2 Network Interface Adapters (NIAs) which form the data link layer and the physical layer. The Ethernet framing protocol, a data link layer protocol, with CSMA/CD access capability is fully implemented on these PCLINK2 cards. The NIA does the transmission and reception of the frames, the data link layer transmission units, independently of the main system (the router system) CPU. This NIA card houses Intel 80186 CPU, Intel 82586 IEEE 802.3 LAN Co-processor and Intel 82501 the serial line component.

The LAN Co-processor programmed to take care of the Ethernet functions of the data link layer. The serial line component does the actual transmission and the reception of the bytes in a frame. And 82186 CPU does the necessary co-ordination between 82586 and 82501. Also it sends an interrupt to the router's main CPU, once a complete frame is received and is ready to be handed over to the IP layer i.e. after the frame has undergone the necessary processing at the data link layer. Interrupt lines IRQ2 and/or IRQ5 (for IBM-AT) of the main system receive this interrupt. The NIA can be jumpered to select either IRQ2 or IRQ5 [37].

A router needs at least two NIAs so that it can connect two LANs. But these two NIAs need not be supporting the same kind of data link protocol. For instance, a router may have two Ethernet NIAs or one Ethernet NIA and one Token Ring NIA or one Token Ring NIA and one PPP serial line adapter or one Ethernet NIA and one PPP serial line adapter and so on. It is also possible that a router has two Ethernet NIAs and one PPP serial line adapter. This is the configuration proposed for the IP router in the ERNET LAB.

5.2 Packet Transaction Using Ethernet NIAs

The figure 5.1 shows the functional blocks involved in the datagram reception and forwarding process of the IITK router. The PCLINK2 NIAs collect the data bytes from the network, form the Ethernet frames and carry out the necessary data link layer processing. Subsequently, once a complete valid datagram is ready the corresponding NIA sends an interrupt to the router CPU.

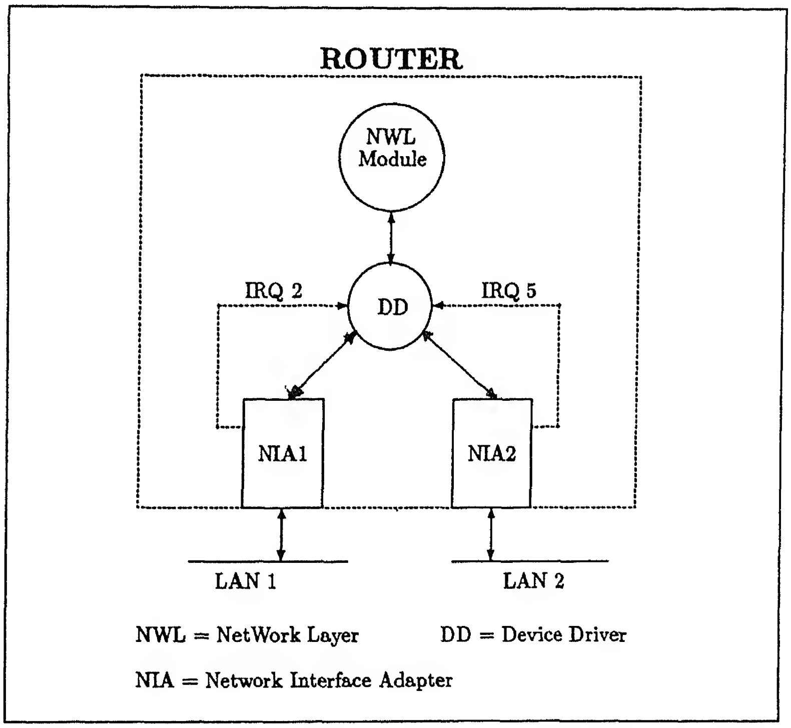


Figure 5.1: Functional Blocks of Router using Ethernet NIAs

The CPU in turn invokes an interrupt service routine defined in the *device driver*[1]. This ISR then collects the datagram from the NIA RAM and hands it over to the IP module for further processing. The IP module, after necessary consistency checkings takes a routing decision based on the destination IP address available in the received datagram and the entries in the *routing table.* Having decided about the NIA the datagram has to be handed over to, the IP module invokes the output module [ch_output( )] defined in the device driver which in turn delivers the datagram to the NIA. The PCLINK2 adapter encapsulates this datagram in the Ethernet frame and transmits on the network.

5.3 Packet Transaction Over Serial Lines

The figure 5.2 shows two possible approaches for packet transaction over the serial lines.

1. Using Serial Line Cards

   and

2. Using the Serial COM Ports.

[1] The basic function and the structure of a device driver is presented in Appendix A.

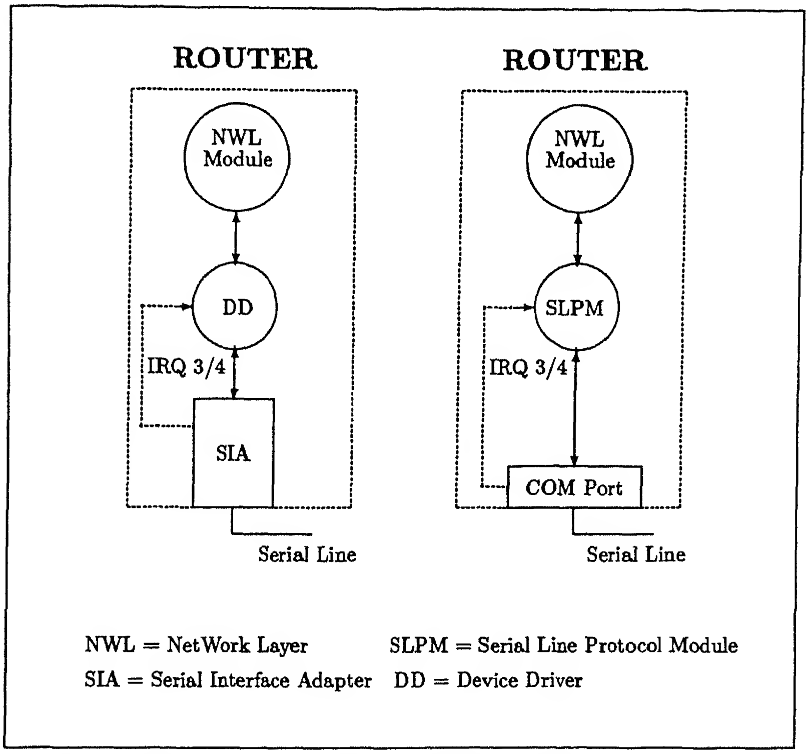


Figure 5.2: Functional Blocks of Router using Serial Lines

In the first case we may have a Serial Interface Adapter (SIA)installed in the router system. This card fully implements the desired serial line protocol, the PPP in our case, and like the PCLINK2 NIA cards, manages the datagram transaction over the line independently of the main CPU. It accomplishes this by using its own CPU. It also sends an interrupt to the main CPU once a complete packet is received and is ready to be handed over to the IP layer (i.e after the datagram has undergone the necessary processing at data link layer). Interrupt line(s) IRQ3 or IRQ4 of the router system are used for this purpose. The I/O routines defined in the device driver are used to deliver the packet to the IP module. Similarly, any packet to be transmitted on the serial line is first delivered to the SIA using the I/O routines in the device driver. After the necessary data link processing the SIA transmits the packet on the serial line.

Alternately, as shown in the second figure, one may use the already existing serial COM ports for this purpose. In this case the main CPU executes the Serial Line Protocol Module (SLPM) and hence the packet transaction is managed by the main CPU only. Whenever an interrupt is recognized on IRQ3 (COM1) or IRQ4 (COM2) lines the Interrupt Service Routine defined in the serial line protocol (PPP in our case) module (now a software module in the router system itself as opposed to that on the serial line card in the previous case) collects the data bytes from the port. Once a complete frame is received, the corresponding data link layer processing is carried out. And if the received frame is correct in all respects it is handed over to the IP module for network layer processing using the I/O handlers provided in the protocol module.

The IP module, after the necessary consistency checkings, takes a routing decision as to which interface the packet should be handed over to. This interface can be one of the two Ethernet interfaces or another serial line interface, if available. In general, for a packet coming from any of the interfaces, if the IP module finds that the packet is to be sent over the serial line, it first hands over the packet to the Serial Line Protocol Module for data link layer processing. The protocol module finally transmits the packet on to the serial line.

It may however be mentioned here that, in this case the router CPU is interrupted for every byte, within a frame, arriving on the link as opposed to the interrupt per frame from the network interface adapters or serial line adapters. This adversely affects the performance of the router.

5.4 PPP-IP Interface Design Considerations

The Point-to-Point Protocol implemented in the ERNET LAB is based on the standards specified in the Request For Comments (RFC) 1331[2]. The source code is written in microsoft C so that it can be integrated and interfaced with the existing IP code which is also written in microsoft C. This implementation of PPP heavily relies on *C Asynch Manager (CAM)*, a serial COM port communications package, for the data transfer over the serial link. The figure 5.3 shows the functional blocks involved in the implementation. As shown in the figure the communication takes place only at the data link layer. Because the PPP module is not interfaced with the network layer IP module. In order to understand the functioning of this PPP implementation, it is necessary to have a clear understanding of the C Asynch Manager operation. The salient features and functions of C Asynch Manager are briefly outlined in Appendix B.

[2] RFC 1548 obsoletes this RFC.

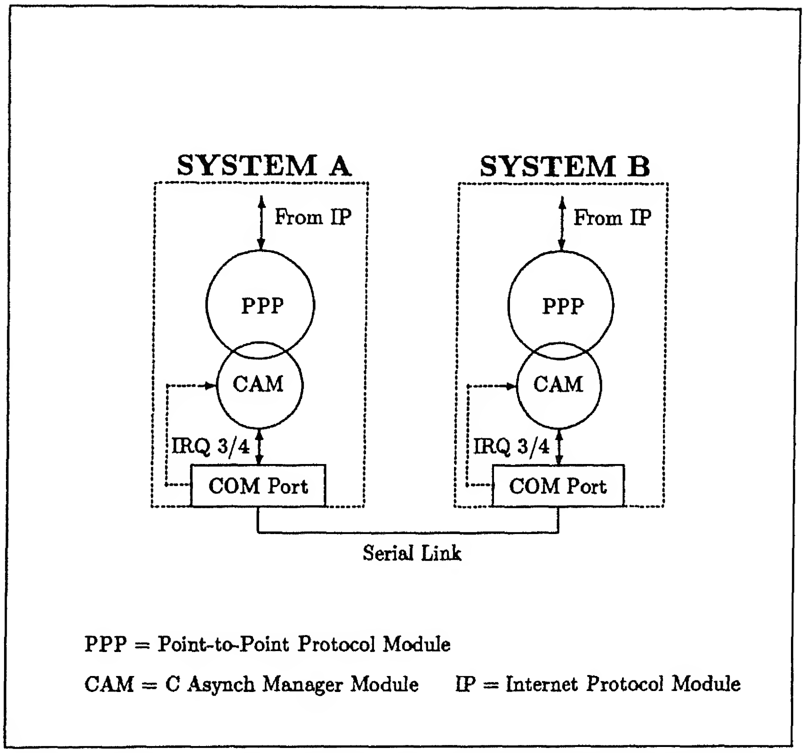

SYSTEM A
SYSTEM B
From IP
From IP
PPP
PPP
CAM
CAM
IRQ 3/4
IRQ 3/4
COM Port
COM Port
Serial Link
PPP = Point-to-Point Protocol Module
CAM = C Asynch Manager Module
IP = Internet Protocol Module

Figure 5.3: Functional Schematic of the PPP Implementation

The PPP accesses the COM ports using the level one C functions defined in the C Asynch Manager. The level one functions use the level zero assembly routines for this purpose.

When the link is to be established a PPP user interface issues[3] an *OPEN* command to the PPP module. In response the PPP module configures the concerned serial COM port using the CAM functions and invokes the LCP module. The LCP module then forms the Configure-Request packet with desired options, transmits over the line and awaits the response from the peer system. If everything is fine the peer system responds with Configure-Ack LCP packet and the two systems negotiate the LCP option. The PPP then enters the IPCP phase and after the necessary negotiation of the IPCP options it sends an *UP* indication to the user interface module. This signals to the user interface module that the datagrams can be transmitted over the serial line. The actions involved in the link establishment phase are shown in the figure 5.4.

[3] A PPP user interface module was developed for the functional verification of the PPP implementation. When the link is to be established this user interface is executed on the two systems interconnected over the RS-232C serial link. It is this interface module which issues the commands to and receives the responses from the PPP module. Theses commands and responses are required to be issued by the network layer, the IP in our case.

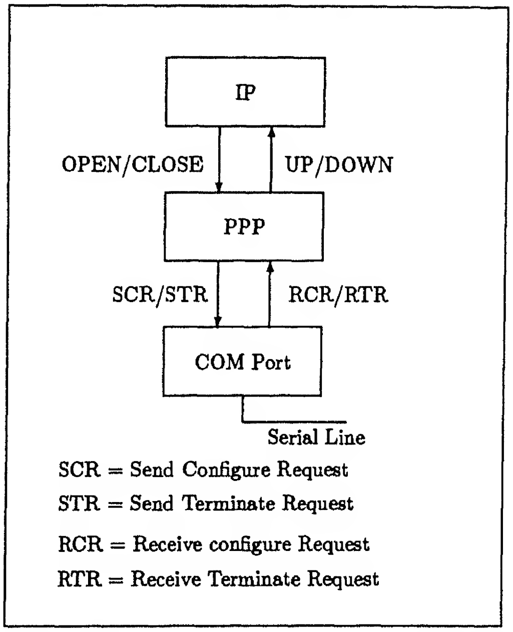

IP
OPEN/CLOSE
UP/DOWN
PPP
SCR/STR
RCR/RTR
COM Port
Serial Line
SCR = Send Configure Request
STR = Send Terminate Request
RCR = Receive configure Request
RTR = Receive Terminate Request

Figure 5.4: Actions involved in the Link establishment Phase

On the other hand if the peer system does not respond[4] and this system times out, PPP closes the COM port and sends a *DOWN* indication to this user interface module. This signals the user interface module that the PPP is in the *link dead phase* and that the link is not available for the data transfer.

Thus the existing PPP module is not able to interact with the IP network module. Nor it is able to run along with the IP router code since there is no scope for the execution of the PPP module once the router module is executing.

As discussed in the section 5.1.1, the IP router code in use at IITK does not support multitasking. But it provides a feature which schedules the the processor among the different processes. Now, in order to run the existing PPP module in association with this router module a process should be defined for the PPP module. This process is the proposed interface for splicing the PPP and IP modules. This interface will have the following three constituents.

- *Functions* for invoking the PPP module for link establishment and PPP data link processing.
- *Functions* for collecting the bytes from CAM input queue

  and

[4] This was accomplished by executing the user interface module on only one of the two systems.

- *Functions* for exchanging the data packets with the IP module.

The implementation of the first two *functions* is explained below.

5.5 Implementation Details

5.5.1 PPP Start Function

A PPP start function is written. This function is invoked from the IP router main function to issue the *OPEN* command to the PPP module for establishing the link at the time of router initialization at boot-up. If the PPP module succeeds in establishing the link with the remote system it sends an *UP* indication to this function. On the other hand if the remote system fails to respond and this system times out, this PPP module leaves the COM port open but sends a *DOWN* indication to the PPP start function. This is an indication that the network layer packets can not be transmitted on the link.

The PPP module is modified to implement this feature. In the earlier implementation the PPP module closes this port if the remote system does not respond before time-out. This detaches the CAM ISR[5] from the COM port interrupt vector IRQ3/IRQ4. As a result it was not possible to receive the data bytes if the remote system tries to establish link with this system at a later instant. In the present implementation the PPP module does not close the COM port. This leaves the CAM ISR attached to the COM port interrupt vector.

From this point onwards the router continues with its normal functions. The CAM ISR collects the bytes, if any, arriving on the serial link and places them in its input circular byte queue. It is now required that these bytes are read from the CAM input queue for PPP processing. This is accomplished by polling the CAM queue.

[5] The C Asynch Manager installs the necessary ISR for handling the interrupt from the COM port when the port is opened. The subroutines defined in this ISR carry out the actual data transmission and reception whenever the interrupt occurs. The C Asynch Manager maintains two (input and output) circular byte queues for storing data. It is this area with which the application program, the PPP in our case, transacts (using the CAM level one C functions) for the data transfer. Thus this storage area is dynamically shared by the application program and the CAM ISR.

5.5.2 CAM Polling Function

This function is periodically executed by the IP router module. It collects the bytes from the CAM input queue. If a complete frame, as indicated by the flags, is received it invokes the corresponding PPP routines for the PPP data link processing. The PPP module after this processing takes an appropriate action. These actions include:

1. Execution of the LCP protocol for link establishment,
2. Sending the *UP* indication to the IP for data transfer,
3. Execution of the IPCP protocol for configuring the IP for PPP

   and

4. Delivering the packet to the IP module (if it is an IP datagram).

The alarm(*time, func, arg, flag*) function[6] defined in the IP router module is used for this purpose. The corresponding router state diagram is shown in the figure 5.5.

Thus this routine enables the router to receive the IP packets over the serial link. But the packets are still at the data link layer only. Necessary I/O routines are required to be written for forwarding these packets to the IP module.

[6] This function schedules an alarm for the execution of a function some time in the future. The arguments have the following meaning.

time

This specifies the time interval after which the function should be executed.

func

This is the function to be executed after this time interval (cam_poll() in our case)

arg

Any parameter to be passed to the function *func*.

flag

This is a repeat flag. It can have two values; *once_only* and *automatic*. The *once_only* flag executes the function *func* only once after the time interval specified by *time*. Where as the *automatic* flag executes this function periodically at the time intervals specified by *time*. For the cam_poll( ) function this flag is set to *automatic*.

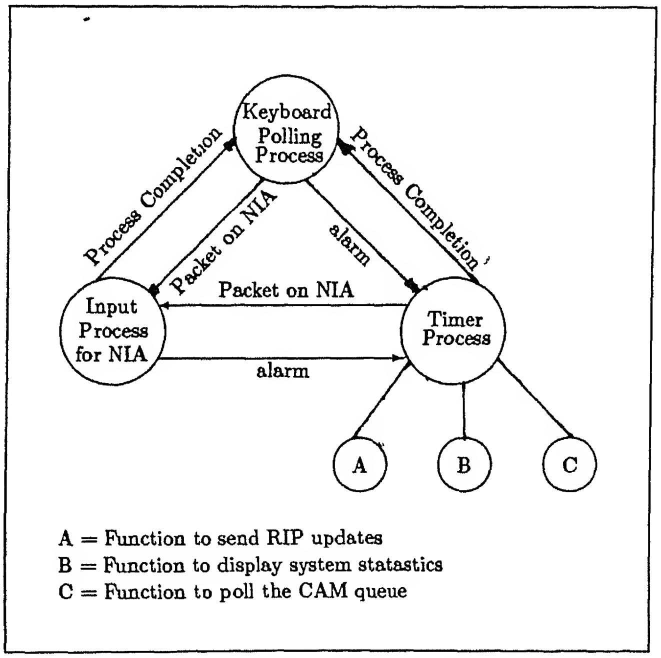


Figure 5.5: Router State Diagram with CAM Polling

5.5.3 CAM Polling Considerations

The rate at which this polling is done should be fast enough so that the circular byte queue does not overflow thereby losing bytes. The input and the output circular byte queues have the maximum capacity of 32K bytes. Let us consider a packet of the maximum transmission length of 1500 bytes (this is the Maximum Transmission Unit in the PPP configuration). This means the queues can accommodate at most 21 full length packets at a time. Further using the *alarm()* function the fastest polling rate can be 1 second i.e. it can cause the PPP to poll the CAM input queue once every second. Rates better than this are not possible to achieve using the *alarm()* function. Another factor to be considered here is the transmission rate of the link. Under the best line conditions one can achieve transmission rates up to 19.2 Kbps using the RS-232C standard. The practical rates are very much below this rate and are in the vicinity of about 9.2Kbps. This means that the polling rate of 1 second is quite sufficient to avoid the loss of data bytes. Additionally the *local flow control* mechanism of the C Asynch Manager module can be enabled to prevent the loss of data.

5.6 Summary

The operation of IITK router is outlined first. Packet transaction using Ethernet NIAs and serial line interfaces is discussed next. Over the serial lines the packet transaction may be accomplished using either a separate serial line card or using the serial COM ports. A function is written to invoke the PPP module for link establishment. The PPP module is modified to configure the COM port for the interrupt driven communication. A polling scheme, using the timer function defined in the IP module, is implemented to receive the data packets over the serial line for PPP data link processing.

Chapter 6

Conclusions and Suggestions

6.1 Conclusions

In this thesis a review and comparison of some of the serial line data link protocols such as HDLC, LAPB, LAPD, LAPF, PPP and SLIP is carried out. From the study it is clear that though all the above mentioned protocols (except SLIP) are derived from the basic HDLC protocol there are some subtle distinctions. For instance, in LAPD and LAPF formats the address field provides multiplexing capability at the data link layer itself. Whereas in LAPB this multiplexing information comes from the X.25 layer 3. This feature enables an end system (DTE) to establish multiple logical connections with several other end systems simultaneously over the single physical link between the end system and the network node (DCE). Though PPP does not provide multiple logical connections it provides a feature for multiplexing the datagrams belonging to different network layer protocols over the same physical link. It defines an additional field, the protocol field, for this purpose in its protocol structure.

SLIP offers a very rudimentary encapsulation support for sending IP datagrams over the asynchronous serial links. It does not support datagrams from other protocols and use of synchronous lines. It does not provide any mechanism for error detection or IP address negotiation.

Table 6.1: A Comparision of Serial Line Protocols

Feature	HDLC	LAPB	LAPD	LAPF	PPP	SLIP
Address Field (Octets)	1/2	1	2	2/3/4	1 (0xFF Broadcast address only)	Not present
Control Field (Octets)	1/2	1	1 for U-frame, 2 for S- and I-frames	Not present	1 (UI packet with P/F bit set to 0. Other values are not defined)	Not present
CRC Code	CCITT-16/ CCITT-32	CCITT-16/ CCITT-32	CCITT-16	CCITT-16	Null-FCS/CCITT-16/ CCITT-32	Not present
ARQ Type	Go-Back N	Go-Back N	Go-Back N	No ARQ	No ARQ	No ARQ
Bit Stuffing/ Destuffing	Supported	Supported	Supported	Supported	supported	Not Suppor
Sequence Numbers	Mod-8/Mod-128	Mod-8/Mod-128	Mod-128	Not Supported	Unnumbered	Not Suppor
Piggy-Backing	Supported	Supported	Supported	Supported	Not Supported	Not Suppor
Maximum Window Size	7/127	7/127	127	Not Applicable	Not Applicable	Not Applica
Information Field Maximum length (octets)	Undefined. Implementations limit it to MTU of NWL	Undefined. Implementations limit it to MTU of NWL	260 for control signalling and MTU of NWL	1600	1500	Not defined
Datagram Multiplexing	Not provided	Provided	Provided	Provided	Provided	Not Provid
Applications	Point-to-Point/ Point-to-Multipoint links	Link level interface for accessing X.25 Network	Link level interface for accessing ISDN Network	Link level interface for accessing Frame relay network	Point-to-point links	Point-to-Po links

NWL = NetWork Layer.

MTU = Maximum Transmission Unit.

The evolution, the protocol structure and the features available in LAPF for the congestion management are studied from the point of implementing a frame relay User-to-Network Interface. The fundamental requirements of a frame relay implementation are the Intelligent End Systems and low error rate digital transmission facilities. An important feature of this protocol is that the routing information, which is normally found in the level 3 of the OSI reference model, is moved down to the level 2. The frame relay nodes operate at level 2 as opposed to the level 3 for X.25 nodes. As a result the processing time at each node is reduced.

Table 6.1 presents a comparison of the studied serial line protocols.

The design issues of a PPP-IP interface are discussed and some of the features of this design are implemented. The PPP module is now able to run in association with the IP router module. The PPP module is modified to support the interrupt driven communication using the COM ports. A polling scheme is implemented so as to enable the IITK router to receive the IP packets over the serial lines. The *timer function* defined in the IP router code is used for this purpose.

6.2 Suggestions for future work

1. The study on frame relay protocol presented in this work may be helpful in the design and development of the proposed frame relay User-to-Network Interface in the Telematics LAB.

2. Using the polling module developed in this work, the PPP frames (and hence the IP packets) can be received. But these packets are not passed to the IP module as the necessary I/O routines are not written for the data exchange between the IP and PPP modules. These routines may be developed. This would enable the router to transact the IP datagrams over the serial links.

3. In the above approach the PPP module is invoked using the timer function defined in the IP module. Instead the PPP module may be invoked by the serial port interrupt. For this, additional assembly routines are required to be inserted in the level zero assembly routines of the C Asynch Manager. The intent is to invoke the PPP module from within the CAM ISR.

Appendix A

Device Drivers

Device drivers provide a consistent software interface to varying network interface devices like Ehternet (PCLINK2 NIA) or Serial Line devices. The intent of a device driver is to hide the device specific details from the upper layer software, the TCP/IP in our case. Each network interface device has one such device driver. Figures A.1 and A.2 help to understand where the device drivers operate in the OSI reference protocol stack. Ethernet and serial line devices are considered.

The IP module recognizes a physical interface below it as an object (software) with a neatly defined structure. This object is an *abstraction* of the actual physical device and contains various

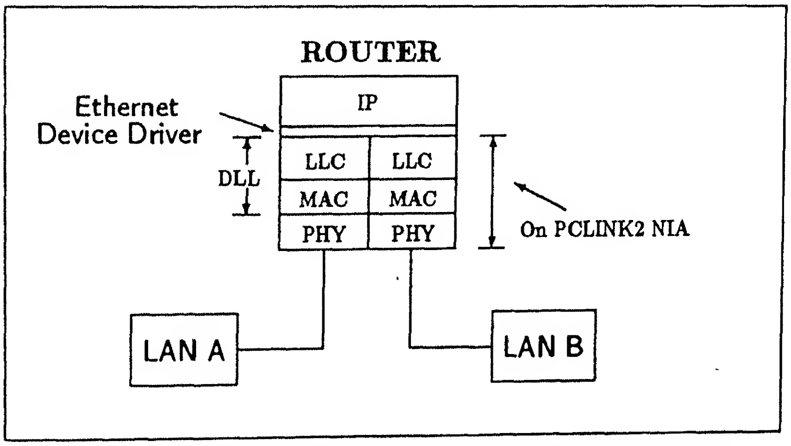


Figure A.1: Router Protocol Stack with Ethernet NIAs

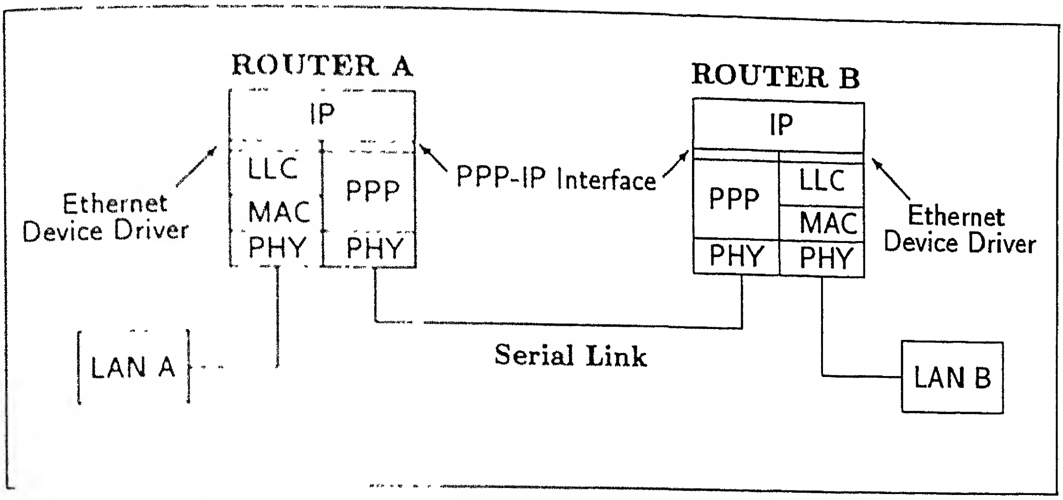


Figure A.2: Router Protocol Stack with PPP Interface

vice specific and device independent parameters. One of the prime functions of a device driver to create this device abstraction and initialize its various parameters so that the IP module can e it for data transmission. This is done at the time of system initialization.

There may be two or more network interface devices of one kind (say Ethernet) installed in a uter system. This is the case when a router connects two LAN segments. In such cases a single vice driver drives all the devices of this category (Like one Ethernet driver driving two or three 'LINK2 cards).

It may be noted that in such cases a single device driver handles the datagrams from two or ree NIAs. Then how the device driver identifies the individual NIAs ? This is done as follows. ı *autoid* structure, defined separately for each NIA, contains two vital parameters, viz *the device ıme* and *the device minor number*. Though the device name (for example *chicken hawk* for CLINK2 device) is the same for the two PCLINK2 NIAs the device minor number, an integer, ıich is *unique* for a particular NIA, distinguishes them. The device minor number is the interface ımber that appears in the routing table. This number enables the IP module to hand over the ıcket to the correct NIA.

The device driver also consists of routines for carrying out the actual data transfer with the ,ers above and below it. These are called Input/Output routines.

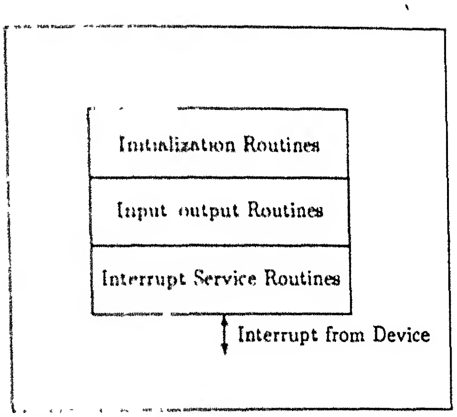


Figure A.3: The Device Driver Structure

A device invokes its driver using the interrupt lines. In order to handle these interrupts the ver also contains suitable Interrupt Service Routines (ISRs).

Thus the device driver routines may be classified in to three categories as shown in the figure 3.

ppendix B

C Asynch Manager

C Asynch Manager is a comprehensive set of routines that can be used in one's C programs order to establish asynchronous communication. Software modules may be written using these utines to drive almost any serial device or communicate with other computers. CAM is designed ecifically to be used with the IBM family of Personal Computers. These routines provide following pabilities.

1. interrupt driven buffered serial communication support
2. support for up to four communication ports simultaneously
3. baud rates of up to 19.2 kbps
4. XON/XOFF flow control protocol support
5. hardware handshaking via RTS/DTR lines of COM ports
6. file transfer capability using XMODEM and YMODEM protocols
7. control functions for *Hayes* compatible modems
8. a layered design isolating low-level assembly language routines from higher level C functions
9. support for Turbo C, Quick C and Microsoft C version 4.0 and later.

The core routines of C Asynch Manager are written in assembly language in order to achieve transmission rates as high as 19.2 kbps

Categories of C Asynch Manager Functions:

The CAM functions may be classified in to five categories as below.

Table B.1: C Asynch Manager Functions

Category	*Description*
Level Zero	Basic asynchronous support
Level One	C interface to level Zero
Level two	Enhancements to level one
File transfer	XMODEM/YMODEM protocol support
Modem control	Hayes modem control routines

The **Level Zero** functions provide the most basic asynchronous communication support. This is the driver package for the COM port. These routines initialize the communication ports, provide interrupt service support, and transfer data between circular queues and communication ports. All Level Zero functions are written in assembly language and drive the *Universal Asynchronous Receiver transmitter* (UART) and the *Programmable Interrupt Controller* (PIC) directly. The level zero functions can be linked directly (as done in the PPP implementation) with the application programs or they may be installed as a memory resident package using the CAM utility *LCOM*. In the former case the functions are accessed via a far call. The CAM library function *comgate* provides this access mechanism.

The **Level One** functions constitute a C interface to level zero eliminating the need to use assembly language. Most level one functions invoke the corresponding level zero routines by using the CAM library function *comgate*. The level one functions are primitives upon which the higher level functions rely. The level one and hence the level zero functions are heavily used in the PPP implementation.

The **Level Two** functions are the general enhancements to level one functions.

The **File Transfer** routines implement the file transfer protocols;XMODEM and YMODEM.

The **Modem COntrol** routines drive standard modem devices. They support Hayes compatible modems directly and can be easily modified to support others.

Configuration of UART:

All IBM Personal Computers provide a dedicated UART chip. The specific chip used is the INS8250 Asynchronous Communication Element. Before any communication chip can be used it must be initialized and programmed. The CAM provides a simple means to do this.

When C Asynch Manager opens a COM port, it performs the following operations.

1. The UART is programmed to enable the generation of interrupts. There are service routines for each of the following four types of interrupts that the UART supports:
   - A character is ready to be read.
   - The transmitter shift register is empty, so data can be written.
   - The modem status has changed.
   - A line error has occurred.
2. An interrupt service routine is installed to handle the four types of interrupts generated by the UART.
3. The level zero routines record the address of a buffer (The application program supplies this address in the *open_a1()* command). The buffer contains the input and output **circular queues** for this port and is maintained by level zero until the port is closed.
4. The logical port numbers are associated with physical devices (address of the UART) and specific interrupt lines.
5. The modem signals Data Terminal Ready (DTR) and Request To Send (RTS) are asserted (although this can be disabled).

The data transmission parameters (baud rate, parity, data bits etc.) can be explicitly set by calling CAM functions.

When C Asynch Manager closes a port the interrupt service routine installed for the port is removed.

Transaction of data with the serial ports:

As mentioned above the level zero routines maintain a separate buffer area, for each port opened, at the address passed to it by the user program using the level one *open* function, open_a1(). The size of the input and output queues, which is specified by the user program at the time of initialization, together determine the buffer size. The total buffer size is limited to 64k, which means that the individual queues may have maximum size of 32k.

To write data to the communication port, if data cannot be immediately transmitted, CAM stores the data in the output queue. When the transmitter holding register is empty (UART is ready to output data), the UART generates an interrupt. The CAM interrupt service routine gains control and transfers one byte of data from the output queue to the UART. In this way the CAM *write* function take care of transmission of the data bytes from the output queue.

Reading data from the communications port is similar. When all the bits of a character are correctly received the UART sets the *receive ready* data flag (in it's line register) and generates an interrupt. Consequently, the CAM level zero *read* routine copies the character from the UART receive data register to the input queue. As long as the input buffer is large enough, no data is lost.

When the application program reads data, the data from the input queue is returned as well as the status of the UART (in case there where any errors such as parity or framing). Data read from the queue by the application program is removed from the queue to make room for newly arriving data.

To summarize, CAM performs the initialization of the UART and supports interrupt processing so that the application programs need not deal with the underlying hardware directly. In other words *the application programs neither read nor write to COM ports directly; rather, they* *either move data from the input queue or to the output queue.*

Transmission Flow Control:

One aspect of asynchronous communication is that either device may transmit data when the receiver is not expecting it. By providing interrupt support and maintaining an input queue for each port, CAM captures data whenever it arrives.

It is possible that a transmitting system may transmit data faster than the receiving system can process it, resulting in a backlog of data in the receiver's input queue. If the backlog grows, it may eventually overflow the receiver's input queue, resulting in data loss. The only way to prevent this is to signal the transmitting system to hold its data until the receiver has processed the previous data. This is called *flow control* or *handshaking* and is supported by CAM in two ways: software and hardware [38].

Bibliography

[1] William Stallings: *Hand Book of Computer Communications Standards, Volume 1*, Macmillan Publishing Company, New York.

[2] TSM.Jagannadha: *Implementation of PPP for IP Router*, March 1993.

[3] Schwartz Mischa: *Telecommunication Networks, Protocols, Modelling and Analysis* Prentice Hall, Englewood Cliffs, New Jersey, 1977.

[4] Antony Rybczynski: "X.25 Interface and End-to-End Virtual Circuit Service Characteristics", *IEEE Transactions on Communications*, Vol. COM-28, No. 4, April 1980.

[5] Sadhiko Kano: "Layers 2 and 3 of ISDN recommendations", *IEEE International Conference on Communications (ICC)*, 1989.

[6] James P. cavanagh: "Applying Frame Relay Interface to Private Networks", *IEEE Communications Magazine*, March 1992.

[7] Simpson W.A.: "The Point-to-Point Protocol for the transmission of multiprotocol datagrams over Point-to-Point links", *RFC 1548*, December 1993.

[8] Andrew Tanenbaum: *Computer Networks*, Prentice Hall of India Private Limited.

[9] Romkey J.: "A non-standard for transmission of IP datagrams over serial lines:SLIP", *RFC 1055*, June 1988.

[10] Xerox Corporation: *Xerox Network Systems Architecture, General Information Manual*, California.

[11] D.E.Carlson: "Bit-Oriented Data Link Control Procedures", *IEEE Transaction on Communications*, Vol. COM-28, no. 4, April 1980.

[12] Ulysis Black: *Data Networks: Concepts, Theory and Practice*, Prentice-Hall, Englewood Cliffs, New Jersey, 07632.

[13] W. Bux, K. Kummerle, H. L. Troung: "Balanced HDLC Procedures: A Performance Analysis", *IEEE transactions on communications*, Vol. COM-28 No. 11, Nov. 1980.

[14] CCITT: " Digital Subscriber Signalling System No.1 (DSS1), Data Link Layer Recommendations, Q.920 - Q.921", *CCITT Blue Book, Vol VI - Fascicle VI.10.*

[15] Simpson W. A.: "PPP in HDLC Framing", *RFC 1549*, December 1993.

[16] Reynolds J. and Postel J.: "Assigned Numbers", *RFC 1347.*

[17] Simpson W. A.: "PPP LCP Extensions", *RFC 1570*,January 1994.

[18] William Stallings: *Hand Book of Computer Communications Standards*, Volume 2, Macmillan Publishing Company,New York.

[19] McGregor G.: " The Internet Protocol Control Protocol (IPCP)", *RFC 1332*, May 1992.

[20] Steven A. Taylor: "Frame Transport Systems" *IEEE Communications Magazine*, March 1992.

[21] Timothy G. Zerbice: "Considering the past and anticipating the future for Private Data Networks", *IEEE Communications Magazine*, March 1992.

[22] Muller, Nathan J.: "Frame Relay, Next Generation X.25 networks", *Journal of Data and Computer Communication*, Summer 1991.

[23] William Stallings: "Faster Packet Networks", *Byte*,November 1991.

[24] M. Irfan Ali: "Frame Relay in Public Networks", *IEEE Communications Magazine*, March 1992.

[25] Moe Rahnema: "Frame Relaying and the Fast Packet Switching Concepts and Issues", *IEEE Network Magazine*, July 1991.

[26] Kim Joanchen, Kelvin K. Y. Ho, Vikram R. Saxena: "Analysis and Design of a Highly Reliable Transport Architecture for ISDN Frame Relay Networks" ,*IEEE JSAC*, Octeber 1989.

[27] Renata Gurneri, Cees J. M. Lanting: " Frame Relaying as a Common Access to N-ISDN and B-ISDN Data Services", *IEEE Communications Magazine*, June 1994.

[28] William Stallings:"Congestion Control in Frame Relay Networks", *Dr. Dobb's Journal*, March 1995.

[29] David W. Petr, Joseph B. Evans, Lyn Neir, Jaswinder Singh, Victor S. Frost: "Access Traffic Control Implementations for Frame Relay", *IEEE JSAC*, 1993.

[30] Paolo Castelli: "Frame Relay Over ATM : Traffic Control Aspects", *IEEE JSAC.*

[31] Wolfgang Fischer, Eugen Wallmeier, Thomas Worster, Simon P. Davis and Andrew Hayter: " Data communication Using ATM : Architectures, Protocols and Resource Management", *IEEE Communications Magazine*, August 1994.

[32] Brett J. Vickers, Tatsuya Suda: "Connectionless Service for Public ATM Networks", *IEEE Communications Magazine*, August 1994.

[33] H. Jonathan Chao, Dipak Ghosal, Debanjan Saha, Satish K. Tripathi: "IP on ATM Local Area Networks", *IEEE Communications Magazine*, August 1994.

[34] Peter Newman: " Traffic Management for ATM Local Area Networks", *IEEE Communications Magazine*, August 1994.

[35] T. Bradly, C. Brown, A. Malis: " Multiprotocol Interconnect Over Frame Relays", *RFC 1490*, July 1993.

[36] William A. Flanagan : "Easing Frame Relay in to the LATA", *Telephone Engineers and Management*, June 1, 1992.

[37] Intel Corporation: *PCLINK2 NIA Hardware Reference Manual*, Intel Corporation, California.

[38] *C Asynch Manager: A Package for Asynchronous Communication*, Blaise Computing Inc.

Glossary

AAL	ATM Adaption Layer
ANSI	American National Standards Institute
ATM	Asynchronous Transfer Mode
BECN	Backward Explicit Congestion Notification
BISYNC	BInary SYNChronous
CCITT	International Telegraph and Telephone Consultative Committee
CIR	Committed Information Rate
CLLM	Consolidated Link Layer Management
CLNP	Connectionless Network Layer Protocol
CPE	Customer Premises Equipment
CRC	Cyclic Redundancy Check
DCE	Data Carrier Equipment
DEC	Digital Equipment Corporation
DLCI	Data Link Connection Identifier
DTE	Data Terminal Equipment
ERNET	Educational Research Network
FCS	Frame Check Sequence
FECN	Forward Explicit Congestion Notification
FRAD	Frame Relay Access Device
FRI	Frame Relay Interface
FRND	Frame Relay Network Device
HDLC	High level Data Link Control
ICMP	Internet Control Message Protocol
IETF	Internet Engineering Task Force
IP	Internet Protocol
IPX	Internet Packet Exchange
ISO	International Standards Organization
ITU	International Telecommunication Union

LAN	: Local Area Network
LAPB	: Link Access Protocol - Balanced
LAPD	: Link Access Protocol - D channel
LAPE	: Link Access Protocol - ISDN Frame Relay
LAPF	: Link Access Protocol - Frame relay
MONOSYNC	: MONO SYNChronous
NIA	: Network Interface Adapter
NLPID	: Network Layer Protocol ID
OSI	: Open Systems Interconnection
PDU	: Protocol Data Unit
PSDN	: Public Switched Data Network
PSTN	: Public Switched Telephone Network
PPP	: Point-to-Point Protocol
PVC	: Permanent Virtual Circuit
SCC	: Serial Communications Controller
SDLC	: Synchronous Data Link Control
SIA	: Serial Interface Adapter
SLIP	: Serial Line IP
SNAP	: SubNetwork Access Protocol
SVC	: Switched Virtual Circuit
TDM	: Time Division Multiplexing
TCP	: Transmission Control Protocol
UNI	: User-to-Network Interface
WAN	: Wide Area Network
XNS	: Xerox Network Systems